BrightStor™ CA-ASM2® Backup and Recovery

Getting Started

4.2

Computer Associates™

Contents

Chapter 1. Introduction

CA-ASM2 is a comprehensive solution for automated storage management. As an integrated, high-performance storage management system, it is designed to increase information center productivity. Productivity is increased by automating all of the functions that control and maintain stored data, while minimizing human intervention. Furthermore, CA-ASM2 positions the information center to manage future storage management requirements in the evolving IBM OS/390 and z/OS operating system environments.

CA-ASM2 performs storage management tasks that include:

- Allocation management
- Backup and restore
- Data set maintenance and volume maintenance
- Movement (migration, archival, and retrieval)
- Reporting
- Cost accounting
- Configuration management
- Disaster recovery

It also provides:

- Interactive facilities for data set maintenance and reporting
- Integration of tape, performance, and security software

Note: CA-RSVP Version 1.0 is a storage management product which is shipped with and operates under CA-ASM2. It is installed as part of the base CA-ASM2 installation.

As user demands for information continue to grow, the requirements for better storage management escalate. Today, information centers are looking for an automated storage management solution that provides the widest range of functionality, while striving to maintain an optimal balance between cost and performance as dictated by organizational and end-user service levels.

1.1 Automated Storage Management Objectives

Storage devices and media have differing characteristics such as space, speed, and cost. The information center needs to match the processing requirements of the information to the device that best achieves the desired results. This is often a balancing of the space-speed-cost and usually is weighted by a service requirement that emphasizes one or more of the issues.

The paramount role of the storage administrator is two-fold: ensure that information is strategically placed in the storage location that best meets performance requirements at any given point in time, and keep resource waste minimal.

As a rule, high-performance devices, such as cached devices with accelerated access speeds and I/O transfer rates, cost more than slower, low-performance devices such as noncached devices. Performance is synonymous with access speed. Information with high-performance requirements need to be placed on faster devices, which in most cases is also costlier.

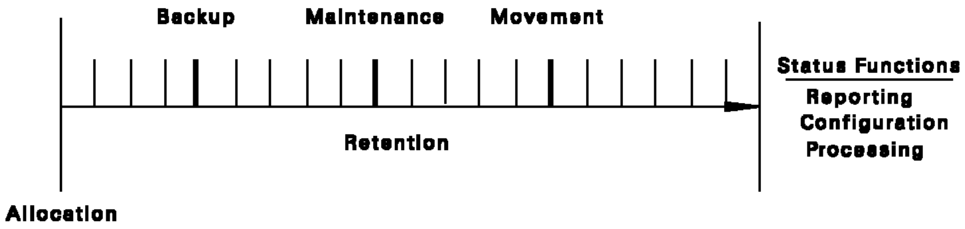
Movement of data from one storage location to another regardless of device type is determined by the processing requirements of the application. Highly active data (data sets in constant demand whose availability is critical) would be reserved for high-performance media. Inactive data sets (those rarely accessed or perhaps not accessed at all) should be offloaded to slower, less costly storage media.

Effective storage management yields tremendous savings. Vast amounts of costly storage space can be freed up by moving information as service needs require. Automating manual processes increases productivity and service to clients in the information center. Proper storage management lets the storage administrator limit storage costs and, also, maintain and control the growth of storage prompted by rising user demand.

1.1.1 Basic Storage Management Functions

The basic underlying principle of automated storage management is the management of information. All information has a life cycle. Basically, a life cycle has a beginning which is allocation and an end which is determined by retention. During the information life cycle, there are certain tasks beyond the normal processing that must be performed at various intervals. These tasks include backing up information, maintaining it, and eventually moving the information to another storage location.

As illustrated in the following graphic, there are five life-cycle related functions. In addition, three more functions relate to status.



Information Life Cycle Functions

Traditionally, these functions have been applied differently to either disk or tape. The reason for this is that the management of different storage media has evolved along the independent characteristics of each. But, the true relationship of these functions applies to data sets, volumes, and devices of all types.

1.1.1.1 Life Cycle Functions

Allocation of a data set begins the life cycle. The primary intent during allocation is to select an appropriate storage location that best matches the processing requirements of the data set. Selecting data set attributes and storage media/device is facilitated by information that can be stored in the Storage Management Subsystem (SMS) data class or storage class, Job Control Language (JCL), or an external knowledge base. Altering allocation information may be performed by allocation management facilities such as Automatic Class Selection (ACS) routines or available equivalents. Effective allocation management at the time of data set creation can reduce or minimize the amount of follow-on tuning.

Frequently, information is updated or changed and must be backed up for contingency recovery purposes. Both data sets and volumes require backup at various intervals during the life cycle. The type of backup is usually based on the type of recovery that may need to be performed. Backup and recovery operations include full-volume and both incremental and explicit for individual data sets. The frequency and timing of backup is often implemented to meet a set of service and recovery needs. Service levels are translated into rules that are stored in parameter libraries or the SMS management class. To achieve the greatest degree of integrity for backup, all processing of the information must cease. This is frequently considered to be disruptive towards meeting the processing needs. Balancing data availability and data integrity issues continues to be a major challenge to most information centers.

Both data sets and storage media may require maintenance operations from time to time to ensure processing efficiency and effective utilization of the storage resource. These operations include reorganization, or even release of excess or overallocated space. Parameter libraries or the SMS management class offer centralized rules control for space release and data removal. Parameter libraries or the SMS storage groups offer rules control for volume reorganization on a scheduled basis.

Another information life cycle function is movement. The movement of data and removable media includes operations such as archival, migration, and vault management throughout a life cycle. Parameter libraries or the SMS management class offers a rules base for managing service levels for DASD data movement. Removable media, such as tape, depends on control files provided by the tape management system for service level control.

Retention identifies the end of the information life cycle. Methods for determining the end of the information life cycle have been traditionally different for DASD data sets than those for tape. Once the criteria have expired, the information may be destroyed and the storage resources freed to receive new allocations.

1.1.1.2 Status Functions

No information center can possibly function efficiently without storage reporting. Storage reporting provides the opportunity to monitor the storage environment, isolate exception situations that require automated storage management, and communicate resource usage to the appropriate affected parties. It is through reporting that a storage administrator can measure storage utilizations and make effective decisions in managing these critical, costly resources.

Physical configuration of storage can limit optimal utilization from both space and performance concerns. Logical configuration of storage volumes, both disk and tape, offers many advantages in improving storage control. SMS controlled storage groups provide for logical grouping of DASD volumes. Dynamic changes to the logical storage configuration allow information centers to adapt to changing processing requirements.

Optimal processing of data regardless of the storage media can be influenced greatly by the processing format. Functions for sequencing, reblocking, and buffer control are available in utility formats and assist in increasing the speed and reducing the cost of data set processing and storage.

These basic storage management functions help information centers ensure data availability, maximize the return on storage investments, meet service level objectives, facilitate disaster recovery needs, and ensure data integrity. These functions also assist in accommodating new requirements related to changing technologies.

1.2 CA-ASM2 Benefits

CA-ASM2 provides a total solution to DASD storage resource management. Today, automation is needed to ensure efficient use of storage resources. The only answer to managing thousands of data sets, hundreds of volumes, tighter schedules, higher service requirements, and increased workloads without additional staff or time is to automate. With CA-ASM2, the information center is provided with a wide range of automated facilities to effectively minimize storage expenditures and ensure optimal throughput and performance.

With CA-ASM2, the storage administrator can implement controls that ensure data is properly placed and secured during its life cycle. Information is available regarding the data through a full-screen online facility that can also be utilized for activating CA-ASM2 data management functions. Unused space can be identified and reclaimed through sophisticated space management facilities.

CA-ASM2 can run as a stand-alone solution, or it can run with other solutions in information centers. It supports all standard IBM data set organizations, including the VSAM Integrated Catalog Facility (ICF) and non-ICF catalogs, SAS, and DB2. It also supports all standard IBM and IBM-compatible direct-access storage devices (DASD), and mixed tape environments.

1.2.1 Improves Data Set Management

CA-ASM2 is designed to optimize data set allocation management and eliminate the lengthy manual processes associated with storage management. DASD performance and management strategies can now be easily implemented to provide enhanced control of data set allocation and DASD resources. These strategies enable standards enforcement of space allocation and usage standards to better manage stored data.

The Allocation Manager component of CA-ASM2 can enforce standards for the allocation of new data sets and provide restricted access to specified data sets, volumes, and units. This facility can direct the allocation to the appropriate DASD pool.

1.2.2 Provides Extensive Management Resources

CA-ASM2 furnishes powerful management reporting facilities. Through a combination of online and batch utilities, CA-ASM2 provides up-to-the-minute information on any DASD volume or data set. Integrated into CA-ASM2 is the powerful, but flexible Report Selection and Variable Processing (CA-RSVP), which is the most comprehensive storage reporting facility available today. CA-ASM2 offers a comprehensive package of DASD management tools that allow enforcement of individual storage management standards and customization of report formats.

1.2.3 Increases Information Center Productivity

CA-ASM2 eliminates the labor-intensive processes typical of storage management systems by automating the selection and placement of data across various, and mixed, storage media. With CA-ASM2, automation and increased productivity are synonymous.

1.2.4 Reduces Costs

The information center realizes substantial savings soon after implementing CA-ASM2 by reducing the costs of three major information center expenses:

- DASD storage
- Personnel to manage stored data
- DASD migration

1.2.4.1 DASD Storage

Overallocated and misallocated data sets are costly trends in information centers today. The costs compound over time as user activity inevitably grows. CA-ASM2 reverses these trends by analyzing data set allocation and usage, and providing the storage administrator with the tools to take corrective action.

Inactive data sets (those not accessed within a predefined period of time) occupy costly online storage. The storage administrator can instruct CA-ASM2 to automatically archive inactive data sets to less expensive media.

Many information centers exclude production data from archive to avoid costly production reruns when data is unavailable. CA-ASM2 offers the Intelligent Transparent Restore (IXR) facility to overcome this problem. IXR automatically reloads production data (and any data archived by CA-ASM2) at either the job or step level. As more data is made eligible for archive, online storage costs are proportionately reduced.

Improving the utilization of existing storage resources in the information center can control the growth of DASD, defer additional hardware expenditures, increase throughput, and provide a greater return on the present hardware investment.

1.2.4.2 Personnel to Manage Stored Data

Controlling DASD growth and user demand for stored data often pushes information center staffing requirements beyond acceptable limits. Yet personnel resources are required to enforce an organization's established storage standards, monitor DASD space allocation and usage, generate reports, continuously perform maintenance and management functions, and develop strategic plans to control DASD growth.

By automating most of these critical storage management functions, CA-ASM2 increases the productivity of the information center's current staff dramatically. This reduces staffing requirements even as DASD growth escalates.

1.2.4.3 DASD Migration

CA-ASM2 has an efficient migration facility to move data between storage locations. Data sets may be moved freely between SMS and non-SMS controlled volumes. Management, data, and storage classes may be reassigned for each data set being migrated to SMS controlled volumes.

Valuable for conversions to new hardware technologies or when moving data for performance reasons, the migration facility is fast and cost-effective. It automatically reblocks data sets to optimum block sizes, and calculates space requirements based on the new block size and the physical characteristics of the receiving device.

1.2.5 Ensures Data Integrity

CA-ASM2 offers a host of facilities that protect an organization's information resources. It provides a comprehensive collection of backup and recovery facilities that ensure that current data is available to the user. Security interfaces with leading security systems, CA-ACF2, CA-Top Secret, and IBM's RACF, ensure that access to data is never compromised, even when moved from another storage medium.

1.2.6 Improves Performance

CA-ASM2 improves system-wide performance several ways. To highlight a few examples, it can:

- Minimize the need for manual intervention by automating the complex processes of storage management.
- Monitor data availability and automatically reload archived data through the Intelligent Transparent Restore (IXR) facility, which runs as a subsystem of the operating system. By ensuring that data is constantly available, the risk of processing delays or job reruns is reduced and system throughput is improved significantly.
- Use efficient data set compression and volume defragmentation facilities to minimize job abends resulting from insufficient DASD space.
- Reposition the Volume Table of Contents (VTOC), on request, near the center of a restored volume, thus accelerating I/O access speed considerably.
- Reposition data sets based on use either by command or by suggested placement from CA-FastDASD.
- Optimize tape usage by retrieval processing and minimize tape mount activity by reloading multiple data sets from a single tape in the order of their occurrence on tape.
- Enforce realtime standards, directing data set allocations to appropriate volume groups or pools.

1.2.7 Expedites Disaster Recovery

CA-ASM2 provides reliable backup and recovery procedures to secure all data sets, and the essential tools for immediate restoration of all volumes and data set information should a catastrophic event occur.

The Stand-Alone Restore facility (SAR) offers smooth recovery from system crashes engendered by, for example, damaging head crashes. This facility allows for the recovery of the system volume even without the operating system functioning. SAR is reliable, straightforward, and easy to use.

For off-site disaster recovery operations, IXR can reload backed up and archived data. It restores or reloads individual data sets, not entire volumes, and only the data sets that are required. IXR controls which archive or backup version to use, selecting the most recent version for each data set.

1.2.8 Designed for Flexibility

The flexibility of CA-ASM2 lets the information center customize storage management functions as required. CA-ASM2 options can be changed at any time without an IPL of the operating system. Flexible runtime parameters permit the storage administrator to tailor specific management functions on a case-by-case basis before they are run. Simulation runs can be executed for planning and testing purposes.

1.2.9 Provides Comprehensive Tape Management

Among its many features, CA-ASM2 provides comprehensive tape management. Tapes can be selected from a preallocated tape pool or a scratch pool. CA-ASM2 optionally controls the selection and recycling of tapes. Interfaces with CA-1 and CA-Dynam/TLMS furnish tape storage control and management.

1.3 CA-ASM2 Operating Environments

CA-ASM2 software supports z/OS and OS/390 operating environments and their equivalents.

The base functions of CA-ASM2 can be installed just SMP/E in a matter of hours without any modifications to the operating system or job entry subsystem (JES). Installing CA-ASM2 with modifications provides additional functionality such as aging, Allocation Manager, and Intelligent Transparent Restore (IXR).

CA-ASM2 is release-independent; all facilities operate without restriction under any installed version of the supported operating systems.

CA-ASM2 is written in IBM 370 Assembler language and does not have to be assembled for execution. It provides several exits that allow tailoring to meet the unique requirements of the information center. However, with CA-RSVP and simple customization options, exits are not normally necessary for CA-ASM2 execution.

1.4 CA-ASM2 User Productivity and Control

CA-ASM2 maximizes user productivity and control. All CA-ASM2 facilities are fully integrated, well defined, and simple to understand. Storage management is simplified so that even end-users can govern their own data sets.

CA-ASM2 user commands are flexible, and can be issued from ISPF panels, as TSO commands, or in batch mode. All CA-ASM2 commands use English-like syntax, making them easy to use, and easy to remember. For example, **$BK DSNAME(mydataset)**. $BK is the backup command and mydataset is the data set to be backed up.

The ISPF command panels offer even greater convenience to all levels of users.

1.4.1 CA-ASM2 ISPF Panel Interface

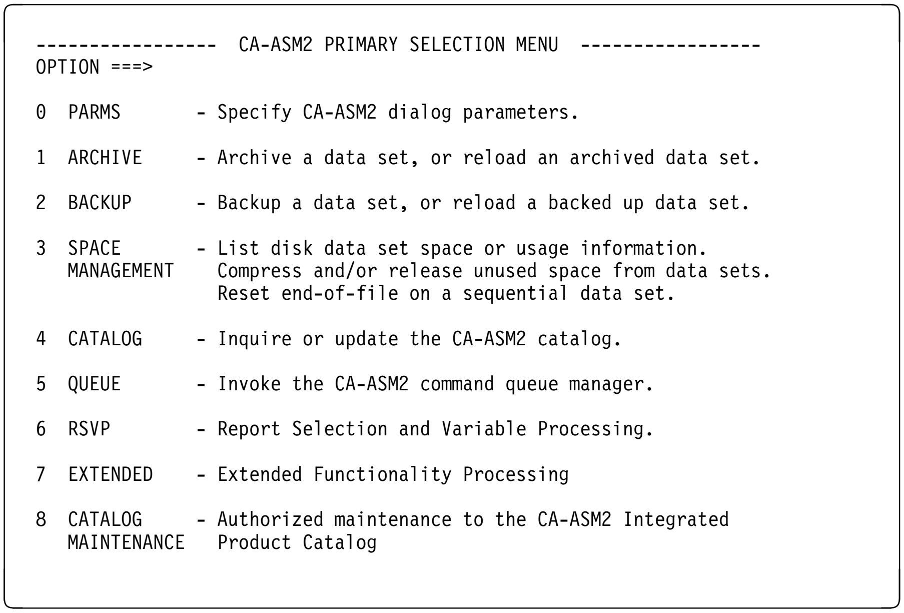
The ISPF panels are integrated into the ISPF/PDF environment under a dialog manager for immediate command execution. Users view full-screen interfaces to CA-ASM2 commands, which familiarize them with command syntax and direct them toward specific command actions.

Users derive several benefits from the ISPF panels, which provide:

- **Tools to accelerate CA-ASM2 training**. Commands and options are arranged visually and logically by command function. Nontechnical personnel and new users quickly master CA-ASM2 facilities.
- **Online tutorial panels**. Background information on CA-ASM2 and concise descriptions of keywords are quickly accessible from any command panel.
- **Easy access to CA-ASM2 commands and options**. Users do not have to remember multilevel command syntax; they simply fill in the blanks to construct each command.
- **Swift entry of commands**. The ISPF panels display the same keyword names that CA-ASM2 uses in online command mode. Users simply enter keyword values in the blanks provided.
- **Immediate syntax error detection**. The dialog manager checks syntax as soon as a command is entered, for faster response time with fewer errors.
- **Browsable output**. All CA-ASM2 output, such as DASD space management reports, backup and archive CA-ASM2 catalog inquiries, and CA-RSVP exception reports can be reviewed using standard ISPF Browse facilities.
- **A Saved Command Table for $RSVP commands**. Users step through a series of parameter entry panels to facilitate building $RSVP commands, which can then be saved to a common library and recalled for use or modification.

The CA-ASM2 Primary Selection Menu under ISPF permits access to all CA-ASM2 command panels.

ISPF Primary Selection Menu



```
-----------------  CA-ASM2 PRIMARY SELECTION MENU  -----------------
OPTION ===>

0  PARMS       - Specify CA-ASM2 dialog parameters.

1  ARCHIVE     - Archive a data set, or reload an archived data set.

2  BACKUP      - Backup a data set, or reload a backed up data set.

3  SPACE       - List disk data set space or usage information.
   MANAGEMENT    Compress and/or release unused space from data sets.
                 Reset end-of-file on a sequential data set.

4  CATALOG     - Inquire or update the CA-ASM2 catalog.

5  QUEUE       - Invoke the CA-ASM2 command queue manager.

6  RSVP        - Report Selection and Variable Processing.

7  EXTENDED    - Extended Functionality Processing

8  CATALOG     - Authorized maintenance to the CA-ASM2 Integrated
   MAINTENANCE   Product Catalog
```

All the CA-ASM2 command panels listed on this menu are shown in the applicable section in this guide.

1.5 Using This Guide

This publication is for users who desire to install Version 4.2 of CA-ASM2 using SMP/E. It is designed for systems programmers and other personnel responsible for the implementation and maintenance of CA-ASM2.

The *CA-ASM2 Planning Guide* outlines information that must be considered prior to installation. Many decisions need to be made to customize CA-ASM2 to operate in your unique environment. It is during this phase that the DASD Storage Administrator or other decision making individual(s) makes these decisions and enters the appropriate parameters on the worksheets provided in the Planning Guide. The worksheets **must** be filled out before actually doing the installation described in this document. With the worksheets filled out ahead of time, the installation proceeds in a timely fashion.

The installation phase is divided into three parts; the base product, IXR, and Allocation Manager installations. All can be installed at once or IXR and Allocation Manager may be installed at a later time. Each of these installations are designed to be easily installed if the associated worksheets have been completed ahead of time. The tailoring phase explains how to fine-tune the system once you have it installed so that it suits your particular requirements.

1.6 Summary of New Features for Version 4.2

Each of the following Version 4.2 enhancements for CA-ASM2 are described in detail in the new set of documentation issued for this version.

1.6.1 9999 Files Tape Support

CA-ASM2 now supports up to 9,999 files on a single CA-ASM2 Archive or Backup tape.

- Allows unload to tape to hold 9999 files
- $FORMAT automatically updates LOxxx file(s)
- Converts old format LOxxx files to new format

1.6.2 SMS Support Enhancements

The Management Class fields controlling expiration may now optionally be used to control when CA-ASM2 will expire, archive and backup versions. In addition, CA-ASM2 invokes the ACS class selection routines prior to reload to handle situations in which the ACS class selection rules have changed and the data set is assigned to different classes.

- Reload pre-drives ACS routines to determine SMS classes during dynamic allocation
- Uses SMS MGTCLASS for retention values during archives and backups

1.6.3 ISPF Interface Enhancements

The CA-ASM2 ISPF application has been enhanced to provide additional functionality to end users of CA-ASM2. Online panels are provided to perform common functions such as requesting the restore of all data sets for an application through a new Application construct and restoring a volume from a specific point in time. Customization capabilities are built in through the use of a profile, which removes options that are inappropriate for general users.

- Group data sets at the application level
- High-level qualifier masking and wild cards

1.6.4 IXR Enhancements

The temporary name created by IXR to restore a data set now supports the specification of up to three alias levels for non-VSAM data sets and two levels for VSAM data sets.

Also a New RLDTMPNM PARMLIB keyword for data set high-level qualifier usage during reload UCAT processing has been added.

1.6.5 CA-ASM2 Workstation

CA-ASM2 integration with the CA-ASM2 Workstation has been enhanced to provide extensive administration, reporting, and storage management capabilities from a Windows-based graphical user interface. CA-ASM2 Workstation is a separately licensed product.

1.6.6 $DASDMNT

Support for SMS management class retention of archived data sets has been implemented via the use of a new keyword $SMSRTPD which can be coded in the SYSIN stream.

1.7 Documentation Changes

- A new document, the *ISPF User Interface Guide* has been added to the documentation set. This document describes all the new extended features developed for Version 4.2. In addition, all ISPF panels that were previously described in Appendix A of the *CA-ASM2 System Reference Guide* and Appendix A of the *CA-ASM2 RSVP User Guide* have been integrated into this document so that a single source can be referenced for all ISPF applications.
- All guides have been updated with relevant Version 4.2 information.
- The syntax throughout the documentation has been given a new look. Information has been provided to assist you in reading the syntax diagrams.
- The Troubleshooting section, found in the *CA-ASM2 System Reference Guide*, now provides information for accessing the Computer Associates home page on the Internet for additional Computer Associates products and services.

1.7.1 Removed

- *CA-ASM2 General Information Guide*. Information located in this guide has been disseminated into the *CA-ASM2 System Reference Guide* and *CA-ASM2 Getting Started*.
- Conversion guides for CA-3 and CA-Dynam/DASD.
- *CA-ASM2 Master Index*.
- Demand Analysis Request (DAR) form. You can now enter your request through StarTCC Extended Support (click on Support at www.ca.com on the Web).

Chapter 2. Base Product Installation

This chapter provides step-by-step instructions for installing the CA-ASM2 base product using SMP/E. Installation of IXR and Allocation Manager are not part of the base installation. You can install the base product, IXR and Allocation Manager all together or install IXR and Allocation Manager at a later date.

Important

It is important that you have thoroughly read the *CA-ASM2 Planning Guide*, specifically "Considerations for Current CA-ASM2 Clients" in Chapter 1, and have completed the three associated worksheets listed next before beginning the installation.

1. Base Installation worksheet
2. $OPTIONS worksheet
3. General Protect Criteria worksheet

Installation of the Intelligent Transparent Restore (IXR) feature of CA-ASM2 is accomplished after installing the base product described in this chapter. The step-by-step procedure for installing IXR is described in Chapter 3, “IXR Installation.”

The Allocation Manager component of CA-ASM2 may be installed at any time after the base product and is described in Chapter 4, “Allocation Manager Installation.”

Caution

Before beginning, check the Program Information Packet (PIP) in the distribution package for any changes to the procedure outlined here.

2.1 Step 1 - Download CAI.SAMPJCL to Disk

Download the ninth file of the CA-ASM2 installation tape to disk by using the following JCL:

```
//jobname  JOB (acct info),CLASS=a,MSGCLASS=x
//IEBCOPY  EXEC PGM=IEBCOPY,REGION=1024K
//SYSPRINT  DD SYSOUT=*
//SAMPIN    DD DSN=CAI.SAMPJCL,
//             DISP=OLD,UNIT=tape,
//             VOL=SER=volser,
//             LABEL=(9,SL,EXPDT=98000)
//SAMPOUT   DD DSN=relpfx.SAMPJCL,
//             DISP=(NEW,CATLG,DELETE),UNIT=unit,
//             VOL=SER=volser,SPACE=(3120,(500,120,90)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PO)
//SYSUT3    DD UNIT=SYSDA,SPACE=(TRK,(5,1))
//SYSUT4    DD UNIT=SYSDA,SPACE=(TRK,(5,1))
//SYSIN     DD *
 COPY INDD=SAMPIN,OUTDD=SAMPOUT
/*
//
```

The above JCL needs to be modified to download the sample jobs needed to complete the installation of CA-ASM2. relpfx is your high-level qualifier for release oriented data sets. Also supply the volser and unit.

2.2 Step 2 - Edit the ISPF L1CHANGE Edit Macro

The ISPF L1CHANGE edit macro is used to change CA-ASM2 variables used during the installation jobs to user-defined names. Most variables that appear in more than one installation task can be specified here and need not be specified for each installation job. You can edit each task separately if you do not have ISPF. Computer Associates recommends that you edit this macro at this time to ensure that the changes to each installation task is uniform,

Note: Variables that are specified only in a single installation task are not modified by this macro. You still need to review the JCL distributed for each task to ensure that any additional modifications are applied before submitting the job.

1. Edit member L1CHANGE of SAMPJCL. Change the TO VALUE in each of the ISREDIT statements.

   Items 1 through 33 and 38 through 41 from your Base Installation worksheet.

2. Save the macro in a CLIST PDS defined to your TSO LOGON procedure.

All of the remaining steps of the installation follow a predefined set of substeps and are as follows:

- Perform each of the steps in the order listed. Some steps have substeps. In these cases, you always pick ONE of the substeps associated with that step based on your unique environment.
- Determine the correct installation task to edit. As an example, if you have a pre-XA system you use one task, if you do not, you use another. All members are in SAMPJCL.
- Edit the task by entering L1CHANGE on the command line. The previously edited ISPF Edit macro then processes that command as though you had entered multiple change commands. If you do not have ISPF or have chosen to edit each task separately, you need to carefully review the JCL for variables.
- Modify any additional variables not modified by the L1CHANGE command. Additional modifications are entered in order from your Base Installation worksheet and are listed in the installation steps.
- Review your changes.
- Submit the job.

2.3 Step 3 - Allocate the Computer Associates Common Target Libraries

If you have already allocated the Computer Associates Common Product Target libraries identified next and verified that they have sufficient space to install CA-ASM2, proceed to Step 5. If the data sets have already been allocated but are not large enough for CA-ASM2, expand these data sets according to the requirements outlined in the *CA-ASM2 Planning Guide* and then proceed to Step 5. If you are installing CA-ASM2 after installing the CA Common Services (formerly Unicenter TNG Framework for OS/390) tape you must modify this job to allocate those data sets required by CA-ASM2 which were not allocated by the CA Common Services components installation. If you have not allocated these data sets or would like to install CA-ASM2 into a separate set of target libraries, you must run this job.

This installation task allocates the Computer Associates Common Product Target libraries (also known as the Permanent Target Files for Computer Associates products). Data sets allocated are as follows:

CAI.CAICICS	CICS modules
CAI.CAICLIB	CLIST library
CAI.CAIISPL	ISPF load library
CAI.CAIISPM	ISPF messages
CAI.CAIISPP	ISPF panels
CAI.CAIISPS	ISPF skeletons
CAI.CAIISPT	ISPF tables
CAI.CAILIB	Load library
CAI.CAILPA	LPA modules library
CAI.CAIMAC	Macros
CAI.CAIPROC	PROCLIB
CAI.CAISRC	Source modules
CAI.PPOPTION	Installation options data set

The data sets above are shown with the default prefix of CAI. This task allocates all Computer Associates Common Target libraries, including those not used by CA-ASM2. Data sets not used by CA-ASM2 are underlined.

Note: The space allocations of the libraries reflect CA-ASM2 requirements only. Data sets not used by CA-ASM2 are given only token allocations.

1. Edit member L142IA1.
2. Type in L1CHANGE on the command line.
3. If you are installing CA-ASM2 after installing the CA Common Services tape, remove the DD statements for those data sets already allocated by the CA Common Services installation. These data sets include CAI.CAILIB, CAI.CAISRC, CAI.CAIPROC and CAI.PPOPTION. You must ensure that these data sets have enough space already allocated to them to hold CA-ASM2, otherwise you must expand them using the guidelines in the *CA-ASM2 Planning Guide* prior to continuing.
4. Review your changes.
5. Submit the job.

2.4 Step 4 - Allocate Computer Associates Common SMP/E Data Sets

New installations which do not have a current Computer Associates common SMP/E environment to install into must perform this step.

This installation task allocates and initializes the Computer Associates Common SMP/E data sets.

1. Edit member L142IA2E.
2. Type in L1CHANGE on the command line.
3. There are no further modifications to this installation task.
4. Review your changes.
5. Submit the job.

2.5 Step 5 - Tailor the CA-ASM2 SMP/E Procedure

This installation task adds the CA-ASM2 SMP/E Procedure to CAI.CAIPROC (or the name you have chosen for CAI.CAIPROC). Other CA-ASM2 procedures are loaded to CAI.CAIPROC in Step 13.

1. Edit member L142IA3E.
2. Type in L1CHANGE on the command line.
3. Review your changes.
4. Submit the job.

2.6 Step 6 - Allocate CA-ASM2 Libraries

This task allocates the CA-ASM2 SMP/E distribution libraries and the CA-ASM2 product data sets which must be unloaded from the distribution tape.

1. Edit member L142IB.
2. Type in L1CHANGE on the command line.
3. Review your changes.
4. Submit the job.

2.7 Step 7 - Unload Data Sets Not Controlled by SMP/E

This task unloads data sets from tape which are not controlled by SMP/E. The CAI.PPOPTION data set and the CA-ASM2 PARMLIB and CA-RSVP sample command files are downloaded by this step. The SAMPJCL data set which is also not controlled by SMP/E was unloaded in Step 1.

1. Edit member L142IC.
2. Type in L1CHANGE on the command line.
3. Review your changes.
4. Submit the job.

2.8 Step 8 - SMP/E RECEIVE

This task does a RECEIVE of the CA-ASM2 base product, IXR and the appropriate version of Allocation Manager.

1. Edit member L142IDE.
2. Type in L1CHANGE on the command line.
3. If you want the base product, IXR and the appropriate version of Allocation Manager, there are no further modifications.

   If you do not want IXR, delete FMID CL142I0.

   If you do not want Allocation Manager, delete FMID CAM42X0.
4. Supply the volser of the installation tape.
5. Review your changes.
6. Submit the job.

Note: If you chose not to install IXR or Allocation Manager by deleting their FMIDs, you can still install them at a later date by executing this step specifying only their FMIDs.

2.9 Step 9 - SMP/E APPLY

This task does an APPLY of the CA-ASM2 base product, IXR and the appropriate version of Allocation Manager.

1. Edit member L142IEE.
2. Type in L1CHANGE on the command line.
3. If you want the base product, IXR and the appropriate version of Allocation Manager, there are no further modifications.

   If you do not want IXR, delete FMID CL142I0.

   If you do not want Allocation Manager, delete FMID CAM42X0.
4. Review your changes.
5. Submit the job. Receiving a return code of 4 is normal.

Note: If you chose not to install IXR or Allocation Manager by deleting their FMIDs, you can still install them at a later date by executing this step specifying only their FMIDs.

2.10 Step 10 - SMP/E ACCEPT

This task does an ACCEPT of the CA-ASM2 base product, IXR and the appropriate version of Allocation Manager.

1. Edit member L142IFE.
2. Type in L1CHANGE on the command line.
3. If you want the base product, IXR and the appropriate version of Allocation Manager, there are no further modifications.

   If you do not want IXR, delete FMID CL142I0.

   If you do not want Allocation Manager, delete FMID CAM42X0.
4. Review your changes.
5. Submit the job.

Note: If you chose not to install IXR or Allocation Manager by deleting their FMIDs, you can still install them at a later date by executing this step specifying only their FMIDs.

2.11 Step 11 - Specify Your $OPTIONS Selections

In this step, specify the CA-ASM2 processing options that you selected for the $OPTIONS worksheet. Be sure to enter only UPPERCASE characters.

Caution

You can modify only the fields in $OPTIONS for which USER MODIFIABLE appears in the comment field. Do not change the length of any field. You can change CA-ASM2 installation options very quickly in the future without reinstalling CA-ASM2 or IPLing the system by simply editing $OPTIONS, then assemble and link edit. As $OPTIONS is dynamically loaded by CA-ASM2 every time it is run, the change takes place immediately except for currently running jobs (IXR). If CAILIB is in your LNKLSTxx, you need to do an LLA refresh after assembling and linking.

CAI.L142.PARMLIB contains all of the user-specifiable processing options and other SYSIN and PARM parameters. $OPTIONS is a member of CAI.L142.PARMLIB.

If you change any of the default settings shipped to you, you need to assemble and link edit $OPTIONS into the CA-ASM2 load module data set (in CAILIB).

If the defaults are satisfactory for the testing phase, you can delay this step. You can modify $OPTIONS at anytime and the changes are picked up with the next invocation of CA-ASM2 functions. See the *CA-ASM2 System Reference Guide* for information on using multiple $OPTIONS files.

After you have completed modifying $OPTIONS according to your $OPTIONS worksheet, (making sure that all entries are in UPPERCASE), perform the following steps:

1. Edit member L142IH.
2. Type in L1CHANGE on the command line.
3. Review your changes.
4. Submit the job.

2.12 Step 12 - Update the Quiet OPEN Table (Optional)

This step is a manual task.

1. Edit the QUIET macro of CAI.PPOPTION and add the names of any programs that you do not want to update the data set aging information. The OPEN and CLOSE modifications use this macro (QUIET). Therefore, if you change this table, you must reinstall or zap the system modifications.
2. Proceed to the next step. This macro is assembled later in the installation process.

Note: The Quiet table is for QUIET READs only. Any data set that is opened for update is aged and the Quiet table is ignored.

2.13 Step 13 - Install CA-ASM2 Procedures

This task loads the procedures to the selected PROCLIB.

1. Edit member L142II.
2. Type in L1CHANGE on the command line.
3. If any global changes to procedures are desired such as changing the default tape unit to cartridge, this is the time to do it. You may do this by entering the following SPF command:

   ```
   C UNIT=TAPE UNIT=CART ALL
   ```

4. Review your changes.
5. Submit the job.

Note: You should define this PROCLIB to JES or you must copy the procedures to a data set that is defined to JES. This must be done before you IPL.

2.14 Step 14 - Allocate and Initialize Data Sets

This task allocates and initializes the CA-ASM2 IPC and control data sets. The control data sets include the queues, tape pool data sets and other data sets which reflect the client environment.

1. Edit member L142IJ.

2. Type in L1CHANGE on the command line.

   Note: The tape pool data sets are allocated and dummy entries are created by this job. At some point you must tailor these. Optionally, this can be done at this time. FREE-VOLSER could be replaced with the actual tape volsers. (FREE-vvvvvv must use FREE- in columns 1-5.)

3. Review your changes.

4. Submit the job. Maximum RC should be 8. (ASM2PUUL in the INITPCTF step always returns RC=8.)

Note: In the event of space abends as a result of submitting this job, you may rerun the task from the top. For VSAM, each step deletes VSAM data sets before creation. For non-VSAM, each step scratches data sets and allocates each individual data set.

2.15 Step 15 - OPEN Modification

CSECT IFG0196W in IBM module IGC0001I:

Prior to this task you **must** have updated the Quiet Macro in CAI.PPOPTION (Step 12).

1. Edit member L142IKE.
2. Type in L1CHANGE on the command line.
3. Modify the SETC variable values &LMOPT, &LMTYP and &PTF in the input CSECT IFG0196W.

   Items 42, 43 and 44 from your Base Installation worksheet.
4. Review your changes.
5. Submit the job.

> **Caution**
>
> Every USERMOD to the system is received and applied but not accepted. Do not attempt to accept these USERMODs.

Note: If you are using another version of IEBGENER (from some other vendor) besides the IBM version, this may result in failure of this job. If this occurs, execute IBM's IEBGENER.

2.16 Step 16 - Authorize CA-ASM2 Commands

Perform Step 16A if you do not have TSO/E.

Perform Step 16B if you do.

2.16.1 16A - Non-TSO/E Users

This task adds CA-ASM2 commands to the TSO Command Authorization Table for TSO versions prior to 1.4.

1. Edit member L142INE.
2. Type in L1CHANGE on the command line.
3. You must modify the SETC variable values &BASE, &EXP and &PTF in the input CSECT IKJEFTE2.

   Items 47, 48 and 49 from your Base Installation worksheet.
4. Review your changes.
5. Submit the job.

2.16.2 16B - TSO/E Users

This task adds CA-ASM2 commands to the TSO Command Authorization Table for TSO Version 1.4 and above.

1. Edit member IKJTSOxx of SYS1.PARMLIB to include the following commands:

   ```
   $RA
   $RB
   $RC
   $RL
   $SM
   ```

2.17 Step 17 - Update Program Properties Table

This manual task updates the operating system Program Properties Table (PPT) to allow CA-ASM2 to perform archive, backup and migration operations on password-protected data sets. Without this update, the system operator must provide the password for each password-protected data set CA-ASM2 is to unload or migrate. The PPT update required by IXR is also included.

Note: Member APPT$ in SAMPJCL contains a detailed write up on updating the PPT.

1. Read member PPTSAMP3 in SAMPJCL. This member contains the commands to be added to the appropriate SCHEDxx member on your system.

2. Edit the appropriate SCHEDxx member to add the commands indicated by PPTSAMP3.

2.18 Step 18 - Modify General Protect Criteria (Optional)

This task saves member ASM2PTAB in CAI.PPOPTION and assembles it to CAI.CAILIB. ASM2PTAB is the Controlled Scratch Protect Table module and contains a sample Protect table. The ASM2PROT macro is used to generate this table.

1. Edit member L142IR.
2. Type in L1CHANGE on the command line.
3. Modify the ASM2PROT macro in accordance with the values you have selected on your General Protect Criteria worksheet. All parameters must end with a comma.
4. Review your changes.
5. Submit the job.

Attention

If you are installing IXR proceed to Chapter 3, "IXR Installation."

If you are not installing IXR but are installing Allocation Manager, proceed to Chapter 4, "Allocation Manager Installation."

If you are only installing the base product, proceed to Chapter 5, "Final Steps."

Chapter 3. IXR Installation

This chapter provides step-by-step instructions for installing the IXR feature of CA-ASM2 using SMP/E. **It is assumed that you have completed the installation of the base product and that you have already installed CAIRIM from the CA Common Services tape.**

To properly implement the Intelligent Transparent Restore feature of CA-ASM2, some planning must be done. It is therefore important that you have read the *CA-ASM2 Planning Guide* and completed the corresponding worksheet entries before following the installation instructions in this chapter.

IXR has intercepts to install at five system exit points.

IEFUJI	(Optional)
IEFUSI	(Optional)
IEFDB401	(Required)
IGG026DU	(Required)
IFG0EX0A	(Required)

Note: Computer Associates recommends using either the IEFUJI or the IEFUSI intercept but not both. Using both is redundant, resulting in unnecessary overhead.

Driver programs are not required at these intercept points unless other exits are already installed there. Detailed information to assist you in driver preparation is discussed in the *CA-ASM2 Planning Guide*.

The IEFUJI and IEFUSI intercepts are installed with CAIRIM, while IGG026DU, IFG0EX0A and IEFDB401 must be installed as SMP/E USERMODs.

Sample SMP/E USERMODs to receive and apply changes to the drivers are also provided in CAI.SAMPLIB. You can customize them to mesh with your system and to invoke the routines invoked by your current exits.

3.1 Step 1 - Install IEFDB401 USERMOD

Note: CAIRIM cannot be used to install this USERMOD.

3.1.1 1A - Without Additional Exit

This task installs the IXR IEFDB401 intercept in SYS1.LPALIB. This procedure should be run using your operating system SMP/E PROC.

1. Edit member CL14241.
2. Type in L1CHANGE on the command line.
3. You must provide the FMID of the owner of IEFDB401 (replace 'XXXXXXX' with the correct FMID in ++VER).

   Item 1 from your IXR Installation worksheet.
4. Verify the SET BOUNDARY statements.
5. Review your changes.
6. Submit the job.

3.1.2 1B - With Additional Exit

This USERMOD creates a new IEFDB401 module in SYS1.LPALIB consisting of the CA-ASM2 driver program, the IXR intercept routine $IXRS99I and an optional, additional IEFDB401 user exit (either another product's or a user written exit).

1. Edit member CL14241B.
2. Type in L1CHANGE on the command line.
3. You must provide the FMID of the owner of IEFDB401 (replace 'XXXXXXX' with the correct FMID in ++VER).

   Item 1 from your IXR Installation worksheet.
4. To include a user-written exit in the IEFDB401, identify the load library where a copy of the module resides (having been processed by the linkage editor).
5. Provide the correct LMODNAME, CSECTNAM and DDNAME where SMP/E can locate the load module.
6. Review your changes.
7. Submit the job.

3.2 Step 2 - Install IGG026DU USERMOD

Note: CAIRIM cannot be used to install this USERMOD.

This task creates a new IGG026DU module in CAI.CAILIB consisting of the IXR intercept $IXR26DU and an additional IGG026DU routine. (Either another product's or a user-written exit.)

Note: Computer Associate's module is required to be first which calls the exit specified. The exit needs to be set up to then call IBM's SVC26.

1. Edit member CL14251B.
2. Type in L1CHANGE on the command line.
3. To include a user-written exit in the IGG026DU, identify the load library where a copy of the module resides (having been processed by the linkage editor).
4. Change all occurrences of CSECTNAM to the CSECT name of your IGG026DU.

   Item 6 from your IXR Installation worksheet.
5. Change all occurrences of LMODNAME to the load module name of your IGG026DU.

   Item 7 from your IXR Installation worksheet.
6. Change the references to DDNAME USERLIB in the SMP/E JCL, the SYSMOD ++JCLIN and ++MOD statements to reflect your DDNAME/DSNAME.

   Item 8 from your IXR Installation worksheet.
7. Review your changes.
8. Submit the job.
9. Edit member CL14252.
10. Type in L1CHANGE on the command line.
11. Provide the owning FMID of IGG026DU.

    Item 5 from your IXR Installation worksheet.
12. Review your changes.
13. Submit the job.

3.3 Step 3 - Install IFG0EX0A

This task receives and applies USERMOD CL14260 to install the IXR missing data set (abend 213-04 condition) intercept. Module L10EX0A is created by this USERMOD in CAI.CAILIB. At IPL time CAIRIM loads L10EX0A under the required name IFG0EX0A.

1. Edit member L142XHE.
2. Type in L1CHANGE on the command line.
3. Provide the owning FMID of IFG0EX0A.

   Item 9 from your IXR Installation worksheet.
4. Review your changes.
5. Submit the job.

3.4 Step 4 - Modify IXR Startup Parameters (Optional)

This is a manual task.

If you have made changes to the default startup parameters on your IXR Startup Parameters worksheet, make these modifications to member IXRPARM of CAI.ASM2.R42.PARMLIB at this time.

Attention

If you are installing Allocation Manager, proceed to Chapter 4, “Allocation Manager Installation.”

If you are not installing Allocation Manager, proceed to Chapter 5, “Final Steps.”

Chapter 4. Allocation Manager Installation

This chapter provides step-by-step instructions for installing the Allocation Manager feature of CA-ASM2 using SMP/E.

The Allocation Manager component of CA-ASM2, including Group Name Modification, can be installed anytime after the base installation has been completed. It is assumed that you have already installed CAIRIM from the CA Common Services tape.

To properly implement Allocation Manager, considerable planning must be done. It is therefore important that you have read the *CA-ASM2 Planning Guide* and completed the Allocation Manager Installation worksheet before following the installation instructions in this chapter.

4.1 Step 1 - ZAP SVC Number

This task applies the SVC number that you have selected for Allocation Manager to selected modules and tables in CAI.CAILIB.

1. Edit member AMIC.
2. Type in L1CHANGE on the command line.
3. Enter the SVC that you have selected.

   Item 1 from your Allocation Manager Installation worksheet.
4. Review your changes.
5. Submit the job.

4.2 Step 2 - Build an Allocation Manager Table

As part of installing Allocation Manager, you must create an Allocation Manager Table (AMT). This table defines the criteria by which jobs are classified and UNIT override values are assigned, as well as other processing options and parameters.

An Allocation Manager Table must be in place when the system is IPLed with the Allocation Manager IEFUJV exit active. To build a default Allocation Manager table which does not perform unit overrides:

1. Edit member AMIK.
2. Type in L1CHANGE on the command line.
3. Review your changes.
4. Submit the job.

See the *CA-ASM2 System Reference Guide* for instructions on how to build an AMT with the Allocation Manager Table utility when you are ready to build your actual Allocation Manager table.

Chapter 5. Final Steps

The final steps to the installation of CA-ASM2 are as follows:

Step A	Define CAIRIM Initialization Parameter
Step B	Assemble the CAIRIM Parameters Table
Step C	Tailor the LMP Keys
Step D	IPL Your System
Step E	File All Materials and Output

These steps are to be followed regardless of which components (base, IXR or Allocation Manager) you have installed.

5.1 Step A - Define CAIRIM Initialization Parameters

This is a manual task.

Each product initialized by CAIRIM must be defined using a PARMLIB member in CAS9, the CAIRIM product initialization procedure supplied by CA Common Services. The PARMLIB DD statement in this procedure points to the PARMLIB member to be updated. As supplied by CA Common Services, this member is named CARIMPRM. The following statement must be added to CARIMPRM to initialize CA-ASM2:

```
PRODUCT(CA-ASM2) VERSION(L142) INIT(L1RIMINT)
```

The entry defined for CA-ASM2 performs installation services for both the IXR and Allocation Manager components of CA-ASM2. Only the specific system modifications identified during the installation are initialized.

The order of product initialization statements in the CARIMPRM member controls the sequence in which SMF exits are called by CAIRIM. For this reason, CA-ASM2 should be one of the last products initialized so that it does not reload data sets (IXR processing) or perform UNIT overrides (Allocation Manager processing) for jobs that may be canceled by other SMF exits. If your installation has CA-7 installed you should place the CA-ASM2 initialization statement before CA-7's so that the CA-7 exit can recognize any jobs canceled by CA-ASM2. User-written SMF exits in SYS1.LPALIB are called after any invoked by CAIRIM.

If CAIRIM has already been installed on your system, ensure that the data set you are installing CA-ASM2 into is defined to the CAIRIM procedure CAS9. All CA-ASM2 modules loaded by CAIRIM are placed in CAI.CAILIB.

5.2 Step B - Assemble the CAIRIM Parameters Table

This task creates the CA-ASM2 L1RIMPRM table in load module format used by the CAIRIM initialization program L1RIMINT. L1RIMPRM is the load module name for the assembled M2@RIMP macro. L1RIMPRM identifies the exact system modifications for IXR and Allocation Manager to be installed with CAIRIM.

1. Edit member L142IW.
2. Type in L1CHANGE on the command line.
3. Modify the parameters specified for the M2@RIMP macro. Enter **Y** for each of the FLG--- flags to indicate which exits were installed with CAIRIM. Enter **N** for those not installed or not installed with CAIRIM. The CA-ASM2 CAIRIM program L1RIMINT uses this assembled table to determine which IXR and Allocation Manager intercepts are to be installed with CAIRIM.
4. Review your changes.
5. Submit the job.

5.3 Step C - Tailor the LMP Keys

The CA LMP execution key provided on the Key Certificate must be added to the CAIRIM parameters to ensure proper initialization of CA-ASM2. To define your CA LMP execution key to the CAIRIM parameters, access the CAS9 procedure supplied by CA Common Services and modify the PDS member pointed to by the KEYS DD statement. The member, KEYS, is supplied during CA Common Services installation and should reside in CAI.PPOPTION, the common options library. The parameter structure for KEYS is described next:

```
PROD(pp) DATE(ddmmmyy) CPU(tttt-mmmm/ssssss) LMPCODE(kkkkkkkkkkkkkkkk)
```

where:

pp Is the two-character product code. **This is required**. (This code agrees with the product code already in use by the CAIRIM initialization parameters for earlier genlevels of CA-ASM2.)

ddmmmyy Is the CA LMP licensing agreement expiration date.

tttt-mmmm Is the CPU type and model (for example, 3090-600) on which CA-ASM2 is to operate. **This is required**.

ssssss Is the serial number of the CPU on which CA-ASM2 is to operate. **This is required**.

kkkkkkkkkkkkkkkk
Is the CA LMP execution key needed to run CA-ASM2. It is provided on your Key Certificate. **This is required**.

Example:

Following is an example of the control statement for the CA LMP execution parameter. The information shown is invalid, and is provided for the purposes of example only.

```
PROD(L1) DATE(15JAN93) CPU(3090-600/370623) LMPCODE(52H2K06130Z7RZD6)
```

For a full description of the procedure for defining the CA LMP execution key to the CAIRIM parameters, see the CA Common Services Services Installation Steps" chapter in *CA Common Services Getting Started*.

5.4 Step D - IPL Your System

It is necessary to IPL your system at this time to install the USERMODs and exits not installed by CAIRIM.

Installation of the base product and IXR is now complete. See Chapter 6, “Tailoring” for information on tailoring of such items as tape interfaces after the installation is complete.

5.5 Step E - File All Materials and Output

Save your installation materials and all output from the installation process. This material is essential for timely and accurate Computer Associates maintenance and support of the product.

Chapter 6. Tailoring

This chapter provides follow-up information and instructions for customizing CA-ASM2. It discusses tape management programs (including interface procedures for various tape management systems), security systems, ISPF support, VSAM list utility, and the Stand-Alone Restore utility.

6.1 Customization Considerations

Follow-up steps are included to tailor CA-ASM2 for your unique environment. Follow the numbered instructions that follow to ensure that key items in your installation are covered. You may reconsider many of these processing options later when you are more familiar with the components and capabilities of CA-ASM2.

1. **Have you established system wide CA-ASM2 archive defaults?**

   - Edit member ARCPARMS of CAI.ASM2.R42.PARMLIB to establish system-wide defaults for $MAXPASS, $MAXTAPE, and $CYCLETM, and to identify valid archivable DASD volumes by $ARCHPAK entries. If there is no $ARCHPAK entry for a volume, CA-ASM2 cannot archive data sets on the volume (unless $ANYPAKS is entered in the SYSIN stream).

     Equivalent SYSIN parameters can override the values for $MAXPASS, $CYCLETM, and $MAXTAPE.

     You can change these parameters at anytime by simply re-editing member ARCPARMS. The changes are effective with the next archive run. Sample values are created as part of the CA-ASM2 installation process.

     The use of $MAXPASS, $CYCLETM, $MAXTAPE, and $ARCHPAK in ARCPARMS is documented in the *CA-ASM2 System Reference Guide*.

2. **Are you permitting any online TSO reloads? If so, have you established an environment in which CA-ASM2 can dynamically allocate tape drives?**

   - Set the value of $MAXCRL in $OPTIONS to limit the number of tape drives that can be tied up by TSO reloads.

   - Establish the correct setting of the bit in the $WTOSW field in $OPTIONS. This controls whether CA-ASM2 should issue a WTOR and await the operator's reply that the operator has the tape reel in-hand before dynamically allocating a tape unit.

   - Check the $USRAUTH field in $OPTIONS. This field controls whether all users, or only those users who already have the MOUNT attribute (according to UADS), can dynamically allocate tape drives through $RA/$RB.

3. **Do you want CA-ASM2 to interface with your tape management system?**

   CA-ASM2 supports interfaces to CA-1 and CA-TLMS. For a detailed description of these interfaces, see 6.2, “Tape Management Support” on page 6-6.

4. **Have you supplied the TSO users with HELP documentation?**

   CA-ASM2 provides HELP documentation for its own commands. The description of their usage applies also to the batch environment. The $HELPCPY job stream in SAMPJCL helps you copy these HELP entries into your own HELP data set.

5. **Do you want CA-ASM2 to interface with your data security system?**

   CA-ASM2 supports interfaces with commercially available data security systems: CA-ACF2, CA-Top Secret and IBM's RACF. See 6.3, “Security Systems Support” on page 6-23. details.

6. **Do you want to use the queued reload facilities of CA-ASM2?**

   If you intend to use queued reload, follow these steps:

   - Read about queued retrieval (reload) in the *CA-ASM2 System Reference Guide*.
   - Decide whether you need the started-task monitor. If you can run reloads at specified times, ignore $RXQUMON. Inspect the queued-reload modules. If you want to run $RXQUMON as an online task, you must modify it to submit jobs. (By default, $RXQUMON attaches $RA to do the reload. You only need to modify $RXQUMON if you want it to submit batch jobs to do the reload.) If you want to submit a $RETREVE job stream from $RXQUMON, insert the code at label SUBMIT in $RXQUMON. Insert some PUTs to an internal reader.

     Queued reload uses the following modules provided in source form in CAI.CAISRC and in load form in CAI.CAILIB:

     $RXQEXIT is called if $OPTIONS is set to do queued reloads.

     $RXQUMON monitors the request queue. This program is designed to be a started task to monitor the queue. If the queue has more than a certain number of requests or if the oldest request is more than a certain number of minutes old, a link to the reload process is done. You do not need this program if you want to run reloads at specified times of the day. But, you may want to improve service somewhat by using this program as a decision step for a subsequent reload.

     $RELDEXT is the reload postprocessing exit. It sends messages to TSO users if the requests came from TSO.

   - If you need to, modify the above modules and assemble and link them into the CA-ASM2 link list library (and CAI.CAILIB).
   - If you need more than the default of approximately 500 records in the reload queue data set, delete ARCH.$RAQUEUE and reallocate a larger version using the $FORMAT utility. Appendix B, “Formatting CA-ASM2 Data Sets” and Appendix C, “Expanding the Queue Manager Files,” located in this guide, describe this process.

- Execute the following JCL located in member $RXQUMON of CAIPROC:

```
//         JOB
//$RXQUMON EXEC    PGM=$RXQUMON,PARM='XXXNNNYYY'
//SYSPRINT DD  SYSOUT=*,
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*  $RAORD REQD IF RESTORES FROM ARCHIVE ARE TO BE DONE
//*  $RBORD REQD IF RESTORES FROM BACKUP ARE TO BE DONE
//*  $RXQUEUE REQD IF ARCHIVE QUEUE IS TO BE ACCESSED
//$RAORD   DD  DSN=&TEMPRA,UNIT=SYSDA,
//             SPACE=(TRK,(1,1)),DISP=(,DELETE,DELETE),
//             DCB=(RECFM=FB,LRECL=260,BLKSIZE=260)
//$RBORD   DD  DSN=&TEMPRB,UNIT=SYSDA,
//             SPACE=(TRK,(1,1)),DISP=(,DELETE,DELETE),
//             DCB=(RECFM=FB,LRECL=260,BLKSIZE=260)
//$RXQUEUE DD  DSN=prefix.ARCH.$RAQUEUE,DISP=SHR
//*  THE FOLLOWING JCL IS NEEDED FOR DATA MOVERS
//AMSOUT   DD  SYSOUT=*
//ASM00000 DD  SYSOUT=*,
//             DCB=(RECFM=FA,LRECL=121,BLKSIZE=121)
```

You can default or specify the run parameters with the EXEC statement or START command (S $RXQUMON,,,xxxnnnyyy). The order of PARM merge is DEFAULT, EXEC PARM, then STARTED TASK PARM, so the later ones replace the earlier ones.

The PARM field is defined as follows:

```
'xxxnnnyyy'
```

where:

xxx Is the sampling period in minutes. If this field is zero, only one check is made and $RXQUMON then terminates. If xxx is nonzero, it samples every xxx minutes until stopped or canceled.

nnn Is the minimum number of queued reload requests in ARCH.$RAQUEUE needed to submit a batch job to perform reload requests.

yyy Is the number of minutes to determine an aged reload request. If the oldest request is over yyy minutes old, submit the job.

The default is PARM=000000000.

7. **Do you need to do any additional formatting?**

CA-ASM2 uses a series of DASD data sets as described in Appendix A, “CA-ASM2 Data Sets.” Some of these are supplied as part of the installation package (for instance, CAI.CAILIB). Others must be allocated and formatted by a special utility ($FORMAT). This is carried out by Step 15 of the base installation. It may be necessary at a later date to increase the space allocated to any of these data sets. See Appendix B, “Formatting CA-ASM2 Data Sets” for information on formatting.

8. **Are you using VSAM Export Controls?**

   You can use the $VSAMEXM field in $OPTIONS to direct CA-ASM2 to use control interval mode (CIMODE) during the export of VSAM ESDS data sets. VSAM's default mode for EXPORT is record mode. CIMODE can be used for backup and/or archive. For IBM's relational database, DB2 (an ESDS), you have to set $VSAMEXM to CIMODE.

9. **If you are JES3, have you made the required modification?**

   JES3 must have job control statements added to its procedure. For detailed information see the subheading titled "Required JES3 Modification" in Chapter 2 of the *CA-ASM2 Planning Guide*.

6.2 Tape Management Support

CA-ASM2 can be used with TMS, and TLMS. TMS refers to the CA-1 Tape Management System. TLMS refers to CA-Dynam/TLMS.

6.2.1 General Information

You can obtain CA-ASM2 tapes from two sources: (1) a commercial tape management system that also manages the tapes, or (2) a preallocated tape pool managed by CA-ASM2.

To use a preallocated pool, you must inform the tape management system that the designated tapes are not scratch tapes under its control. Typically this means defining the tapes to be used with permanent retention. CA-ASM2 then controls their usage and recycles them as data expires.

The most basic decision to make regarding CA-ASM2 tapes is whether to keep the tapes under control of your tape management system.

Note: CA-ASM2 does not provide TLMS EDM control. You must consult TLMS for EDM support.

6.2.2 CA-ASM2 Tape Pool

CA-ASM2 uses two methods of tape handling. The $TAPPOOL option setting in $OPTIONS determines which method CA-ASM2 uses.

If you select the $TAPPOOL option, CA-ASM2 controls tape usage and cycling. Any tape mounts are made as specific requests. The volsers of the tapes to be used are identified to CA-ASM2 in three data sets:

ARCH.$TAPPOOL	Archive master tapes
ARCH.$DUPPOOL	Archive duplicate tapes
BKUP.$TAPPOOL	Backup master and duplicate tapes

Volsers are first entered by you and controlled by CA-ASM2 from then on.

If you do not select the $TAPPOOL option, tapes are called for without specifying a volser (which results in a mount request for SCRTCH or PRIVAT).

6.2.3 CA-1 (TMS)

CA-ASM2 is compatible with CA-1 without modification to CA-1. Three methods of using CA-ASM2 in a CA-1 environment are listed next. The advantages and disadvantages of each method and step-by-step procedures for interfacing are described next.

The three methods to interface CA-ASM2 and CA-1 are:

1. Both CA-1 and CA-ASM2 control the tapes. A CA-ASM2 tape pool is used and the tapes are recorded in the Tape Management Catalog described later in this chapter.
2. CA-1 controls the tapes with CA-ASM2 freeing the tapes. There is no CA-ASM2 tape pool, but CA-ASM2 still controls tape retention described later in this chapter.
3. CA-ASM2 controls the tapes. There is a CA-ASM2 tape pool; tapes are not recorded in the Tape Management Catalog. For more information see 6.2.3.3, “CA-ASM2 Controls Tapes” later in this chapter.

A dummy data set is written at the front of every CA-ASM2 tape so the installation can provide a meaningful data set name to associate with each archive, backup, or duplex tape. With these methods, the dummy data set is the only data set on those tapes that is included in the Tape Management Catalog (TMC). (Its EXPDT is set with $OPTIONS or the TAPE1 DD statement in UNLOAD30 steps.) Files 2 through n are created with RETPD=0, bypassing the TMC update. This avoids cluttering the TMC with unnecessary data set name entries for the same volume.

6.2.3.1 Both CA-1 and CA-ASM2 Control the Tapes

The advantages to using this method are:

- Full tape protection offered by CA-1.
- Provides full tape recycling capabilities (both master and duplex).
- CA-ASM2 tapes can be kept separately near the tape drive area (critical for online reloads).
- CA-ASM2 tapes can be new tapes. This reduces the chance of physical I/O errors.

The disadvantages to using this method are:

- Continual management and maintenance of the tape pools.
- If no tapes are available in the tape pool, a mount SCRTCH or PRIVAT is issued resulting in tape active but not in the tape pool.
- To solve above problems, overallocation of tapes in tape pool causes wasted tape media resources.
- Manual updates of the TMC for each volser in the tape pools.

To use this method, follow this step-by-step procedure:

1. **Update $XTTMS in CAI.CAIMAC.**
   - First, indicate you want a CA-1 interface by setting the variable &TAPEMGT accordingly:

     ```
     &TAPEMGT SETC  'TMS'    TMS INTERFACE
     ```

   Note: If you are installing Version 5.1 or later of CA-1 use 'CA1'.

- Recall the prefix you chose for CA-ASM2 data set names. Use the same prefix for the data set name written to the first file of every CA-ASM2 tape as a tape ID. At numerous points in the CA-ASM2 install procedure, you decided whether to choose a prefix. If you chose a prefix one place, you should have chosen the same one everywhere else. But if you decided not to choose your own prefix, the result in other parts of CA-ASM2 differs from the result here. For instance, CA-ASM2 system data sets have names that start with ARCH or BKUP, but the default prefix for tape IDs is set (in the &PREFX statement) to ASM2:

  ```
  &PREFX   SETC  'ASM2'    DSN PREFIX
  ```

  To choose a different prefix for tape IDs, just replace ASM2 in this statement with the prefix you want.

  Note: Pick one naming scheme to use in $NTEXIT and use it consistently. Do not, for instance, change $NTEXIT with each version of CA-ASM2, making tape name release dependent.

- Next, set the value for &TAPPOOL to 'YES':

  ```
  &TAPPOOL SETC  'YES'     CA-ASM2 TAPE POOL USED
  ```

- Set the value of &DUPEDSN to 'YES' to insert the master tape's volser into duplex tape data set names, as follows:

  ```
  &DUPEDSN SETC  'YES'     INSERT MASTERVOL INTO DUPE DSN
  ```

  $XTTMS is now updated. To complete step one, save $XTTMS into CAI.CAIMAC with the changes you made.

2. **Use the CA-ASM2 tape pool and CA-ASM2 exits to update the TMC. Verify or change certain values in $OPTIONS to facilitate this. Review $OPTIONS:**

   $TAPPOOL should be set to X'01'.
   $ARCHEXP should be set to 99365.
   $BKUPEXP should be set to 99365.
   $MNTOPT should be set to X'01'.
   $DUP#MAX should be specified.
   $DUPTMAX should be specified.

   The reason $MNTOPT should be set to X'01' is so CA-ASM2 automatically frees expired archive tapes. Backup tapes are automatically freed. For more details, see the explanation of $OPTIONS field $MNTOPT, located in the *CA-ASM2 Planning Guide*.

3. **Assemble the CA-ASM2 TMS interface exits.**

   You can find sample JCL to do this in member ASMTMS of SAMPJCL. For the assembly step, you must specify three MACLIBs:

   CAI.CAIMAC to include CA-1 and CA-ASM2 macros
   SYS1.MACLIB to include IBM macros
   SYS1.AMODGEN to include IBM macros

These may not be their true data set names in your installation. Check with the person responsible for CA-1 to verify into which MACLIB the CA-1 macros were installed. Prior to Version 4.9 CA-1 macros were installed into UCC1.MACLIB.

- Assemble member $BTTMS of CAI.CAISRC and link it to CAI.CAILIB with the member name $BTTMS. It is automatically assigned aliases of $BT and $BTAPE.
- Assemble member $FTEXIT and link it to CAI.CAILIB with the member name $FTEXIT. It is automatically assigned aliases of $FT and $FTTMS.
- Assemble member $NTEXIT and link it to CAI.CAILIB with the member name $NTEXIT. It is automatically assigned the alias $NT.
- Assemble member $TSINQ and link it to CAI.CAILIB as member $TSINQ.

You have now completed assembling.

4. **Define the CA-ASM2 tapes to TMS. Then initialize the physical tape volumes.**

The tapes **cannot** be marked as scratch or delete. They should have expiration dates of 99365, use counts of zero, and data set names that begin with the prefix defined in $XTTMS.

Each entry in the tape management catalog can have either a complete or partial data set name. If you enter only the highest level of the name (the prefix), you must include the period at the end of the level.

5. **If you are vaulting the CA-ASM2 duplex tapes, make sure you set &DUPEDSN to YES.**

Define in the CA-1 vault pattern data set a vault pattern for each possible CA-ASM2 duplex tape. Specify the vault the tape is kept in with a vault retention of one cycle and have the last cycle kept in the vault until the tape expires. For example in CA-1 prior to Version 5.1:

```
DSN=ASM2.ARCH.DUPE,INDEX,SEPDSN
V=VLT1,C=1,EXP

D=ASM2.ARCH.DUPE.ASM001
V=VLT1,C=1
```

This example designates an archive tape. If you are going to create and vault duplex tapes in the CA-ASM2 backup process, you must establish similar patterns for duplex backup volumes.

You have now completed implementing the CA-ASM2 TMS interface allowing both CA-1 and CA-ASM2 to control the tapes and use the CA-ASM2 tape pool.

If you are CA-1 Version 5.1, code the equivalent 5.1 definitions:

```
DSN=ASM2.ARCH.DUPE
```

6. **Review the tape duplexing module.**

See $RSUCA1 in CAIMAC for CA-1 Version 4.x or $RSUCA15 for CA-1 Version 5.1 or greater for information on maintaining duplex tapes.

6.2.3.2 CA-1 Controls the Tapes with CA-ASM2 Freeing the Tapes

Computer Associates recommends this method be incorporated.

This method provides the full tape protection offered by CA-1. In addition, the master tapes remain CA-ASM2 tapes (EXPDT=99365) until CA-ASM2 frees them without your manual intervention. The disadvantage of this method is that you cannot totally automate the release of duplex tapes from CA-1.

To use this method, follow this step-by-step procedure:

1. **Update $XTTMS in CAI.CAIMAC**

   - First, indicate that you want a CA-1 interface by setting the variable &TAPEMGT accordingly:

     ```
     &TAPEMGT SETC  'TMS'    TMS INTERFACE
     ```

     Note: If you are installing Version 5.1 or later of CA-1 use 'TMS5'.

   - ASM2 can be set up as a CA-1 EDM. Perform the following steps to do this:

     a. Make sure you are at least at ASM2 4.1 Service Pack 2.

     b. Update the TMOEDMxx members with entries like the following:

     ```
     EDM=ARCH,DSN=-,PGM=$MAINT
     EDM=ARCH,DSN=-,PGM=$DASDMNT
     ```

   - Recall what prefix, if any, you chose for CA-ASM2 data set names. Use the same prefix for the data set name written to the first file of every CA-ASM2 tape as a tape ID. At numerous points in the CA-ASM2 install procedure, you decided whether to choose a prefix. If you chose a prefix one place, you should have chosen the same one everywhere else. But if you decided not to choose your own prefix, the result in other parts of CA-ASM2 differ from the result here. For instance, CA-ASM2 system data sets have names that start with ARCH or BKUP, but the default prefix for tape IDs is set (in the &PREFX statement) to ASM2:

     ```
     &PREFX   SETC  'ASM2'    DSN PREFIX
     ```

     To choose a different prefix for tape IDs, just replace ASM2 in this statement with the prefix you want.

     Note: Pick one naming scheme to use in $NTEXIT and stick with it. Do not, for instance, change $NTEXIT with each version of CA-ASM2, making tape name version dependent.

   - Next, set the value for &TAPPOOL to 'NO':

     ```
     &TAPPOOL SETC  'NO'     CA-ASM2 TAPE POOL NOT USED
     ```

   - Place the volser of the CA-ASM2 master tape in the DSNAME of the duplex tapes created for it. This is essential since this method does not use the CA-ASM2 tape pool function. Set the value of &DUPEDSN to YES to insert the master tape's volser into duplex tape data set names. This causes the duplicate tape data set name to be ASM2.ARCH.DUPE.Tnnnnnn where nnnnnn is the volser of the master tape. For example:

```
&DUPEDSN SETC 'YES' INSERT MASTERVOL INTO DUPE DSN
```

$XTTMS has now been updated. To complete step 1, save $XTTMS into CAI.CAIMAC with the changes you made.

2. **Force CA-ASM2 to use CA-1 scratch tape and CA-ASM2 exits to update the TMC as necessary.**

   - Verify or change certain values in $OPTIONS to facilitate this. Review $OPTIONS:

     $TAPPOOL should be set to X'00'.
     $ARCHEXP should be set to 99365.
     $BKUPEXP should be set to 99365.
     $MNTOPT should be set to X'01'.
     $DUP#MAX should be specified.
     $DUPTMAX should be specified.

     The reason $MNTOPT should be set to X'01' is so CA-ASM2 automatically frees expired archive tapes. For more details, see the explanation of $OPTIONS field $MNTOPT, located in the *CA-ASM2 Planning Guide*.

3. **ASM2 can be set up as a CA-1 EDM. To do this:**

   - Make sure you are at least at ASM2 4.1 Service Pack 2.
   - Update the TMOEDMxx members with entries like:

     ```
     EDM=ARCH,DSN=-,PGM,=$MAINT
     EDM=ARCH,DSN=-,PGM,=$DASDMNT
     ```

   - Restart CA-1 to pick up the changes.

4. **Assemble the CA-ASM2 TMS interface exits.**

   You can find sample JCL to do this in member ASMTMS of SAMPJCL.

   For the assembly step, you must specify three MACLIBs:

   CAI.CAIMAC to include CA-1 and CA-ASM2 macros
   SYS1.MACLIB to include IBM macros
   SYS1.AMODGEN to include IBM macros

   These may not be their true data set names in your installation. Check with the person responsible for CA-1 to verify into which MACLIB the CA-1 macros were installed. Prior to Version 4.9 CA-1 macros were installed into UCC1.MACLIB.

   The job ASMTMS:

   - In CAI.CAISRC, assembles member $BTTMS and link it to CAI.CAILIB with the member name $BTTMS. It is automatically assigned aliases of $BT and $BTAPE.
   - In CAI.CAISRC, assembles member $FTEXIT and link it to CAI.CAILIB with the member name $FTEXIT. It is automatically assigned aliases of $FT and $FTTMS.

- In CAI.CAISRC, assembles member $NTEXIT and link it to CAI.CAILIB with the member name $NTEXIT. It is automatically assigned the alias $NT.
- In CAI.CAISRC, assembles member $TSINQ and link it to CAI.CAILIB with the member name $TSINQ.

  You have now completed assembling.

5. **Change the CA-ASM2 JCL procedures.**

   - Change the CA-ASM2 archive PROCs to handle the process for expiring duplex tapes. In each PROC containing a duplex step (TPBKUP70 in ASM2SYSA, ASM2EXPA, and ASM2EXPP), modify the OUTAPE DD statement to include EXPDT=99001 in the LABEL parameter. This causes CA-1 to keep the most recent duplex tape of any master tape.
   - If you plan to duplex backup tapes and archive tapes, change the CA-ASM2 backup procedures to handle the process of expiring duplex tapes. In each procedure containing the duplex step (TPBKUP70 in ASM2SYSB and ASM2EXPB), you can either (1) set up a manual procedure and use EXPDT=99001 for OUTTAPE in the TPBKUP70 step, or (2) assign duplex tapes a retention period equal to the maximum retention for a backed up data set. The drawback of using the retention period is that most duplex tapes are kept longer than necessary.

     CA-ASM2 backup master tapes are automatically purged from CA-ASM2 and CA-1. You receive a message that the tape is purged. When you receive the message, have your tape librarian ensure that the associated duplex tape has expired.

6. **If you are vaulting the CA-ASM2 duplex tapes, make sure you set &DUPEDSN to YES.**

   Define in the CA-1 vault pattern data set a vault pattern for each possible CA-ASM2 duplex tape. Specify the vault the tape is kept in with a vault retention of one cycle and have the last cycle kept in the vault until the tape expires. For example in CA-1 prior to Version 5.1:

   ```
   DSN=ASM2.ARCH.DUPE,INDEX,SEPDSN
   V=VLT1,C=1,EXP

   D=ASM2.ARCH.DUPE.ASM001
   V=VLT1,C=1
   ```

   This example designates an archive tape. If you are going to create and vault duplex tapes in the CA-ASM2 backup process, you must establish similar patterns for duplex backup volumes.

   You have now completed implementing the CA-ASM2 TMS interface allowing both CA-1 and CA-ASM2 to control the tapes and use the CA-ASM2 tape pool.

6.2.3.3 CA-ASM2 Controls Tapes

This method uses only the CA-ASM2 tape pool to control tapes and Computer Associates definitely does **not** recommend it. If you use it, you lose the tape protection offered by CA-1.

To use this method, follow this procedure:

1. Verify or change certain values in $OPTIONS to use CA-ASM2 tapes in the CA-1 environment. Review $OPTIONS:

   $TAPPOOL should be set to X'01'.
   $ARCHEXP should be set to 98000.
   $BKUPEXP should be set to 98000.
   $MNTOPT should be set to X'01'.
   $DUP#MAX should be specified.
   $DUPTMAX should be specified.

   The reason $MNTOPT should be set to X'01' is so CA-ASM2 automatically frees expired archive tapes. For more details, refer to the explanation of $OPTIONS field $MNTOPT, located in the *CA-ASM2 Planning Guide*.

2. Ensure that the volsers assigned to CA-ASM2 tapes in the tape pools do not match any volsers defined to CA-1 even if the latter are defined to CA-1 as in DELETE status. If there is an overlap, operators get a message verifying the tape is from an outside tape library for every data set that gets written to tape. Check with the CA-1 systems programmer on volser ranges in TMS.

6.2.4 CA-Dynam/TLMS (TLMS)

You can use TLMS with or without the CA-ASM2 tape pool option, but you must consider these items:

1. You can place tapes completely under the control of CA-ASM2 and exclude them from TLMS by selecting the $TAPPOOL option and using a volume serial number range outside of the range controlled by TLMS. However, Computer Associates recommends that you not exclude CA-ASM2 tapes from TLMS since TLMS adds a level of integrity not available with CA-ASM2 alone. Accidental destruction of a single CA-ASM2 tape can result in hundreds of lost data sets.

2. Using the $TAPPOOL option of CA-ASM2 has several advantages. It makes CA-ASM2 tapes easy to identify for special care and quick access. And, since you can use a predesignated volume number series, perhaps with a special location, misplaced tapes are less of a problem.

3. CA-ASM2 modifies the TLMS handling of volumes containing multiple data sets so that only the first data set on the volume is updated in the TLMS master file. All other files on the tape are ignored.

4. TLMS data set protection prevents tapes from being overwritten unless the tapes are scratched by a TLMS batch run or by the tape librarian. Without some interface between TLMS and CA-ASM2, there is no easy method to notify TLMS when a tape is no longer needed by CA-ASM2. Computer Associates does not

recommend retention period alone since it is not possible to predict a single expiration date at the time the first data set is written to a volume. Using the system catalog is not effective unless all tapes somehow have unique names. This is possible if you are using the tape pool, but is more cumbersome if you are using scratch tapes.

5. There are three basic methods for CA-ASM2 and TLMS to interact in the freeing of tapes:

   - With $XTTMS variable &TAPEMGT set to TLMS2, assemble the distributed $FTEXIT program to get a $FTEXIT in which the TLMS master file is updated to indicate that the volume being freed is scratched. This approach guarantees synchronization between CA-ASM2 and TLMS and is the default method supplied in $FTEXIT.

   - As an alternative to the above, code a $FTEXIT exit in which a transaction file of scratch volume commands is created. These can be processed by a later TLMS batch run to scratch the volume. If you are using the tape pool option, this approach has the potential problem of CA-ASM2 reusing a tape before the TLMS master file is updated. The exit should set return code 4 to prevent CA-ASM2 from marking the tape for reuse. You can then (manually or by a user program) mark tapes for REUS in the tape pool data set after they are scratched by TLMS.

   - As another approach, place the CA-ASM2 tapes under catalog control in TLMS. The $FTEXIT can then determine the DSN by a call to $NTEXIT or by reading the TLMS master file, and uncatalog the tape. Later batch TLMS processing can scratch the tape. If you are using the CA-ASM2 tape pool, you should make similar provisions as discussed for the transaction file approach.

CA-ASM2 is compatible with TLMS without modification to TLMS. Three methods of using CA-ASM2 in a TLMS environment are listed next. The advantages and disadvantages of each method and step-by-step procedures for interfacing are described next.

The three methods to interface CA-ASM2 and TLMS are:

1. Both TLMS and CA-ASM2 control the tapes. A CA-ASM2 tape pool is used and the tapes are recorded in the Volume Master File described later in this chapter.
2. TLMS controls the tapes with CA-ASM2 freeing the tapes. There is no CA-ASM2 tape pool, but CA-ASM2 still controls tape retention described later in this chapter.
3. CA-ASM2 controls the tapes. There is a CA-ASM2 tape pool; tapes are not recorded in the Volume Master File described later in this chapter.

A dummy data set is written at the front of every CA-ASM2 tape so the installation can provide a meaningful data set name to associate with each archive, backup, or duplex tape. With these methods, the dummy data set is the only data set on those tapes that is included in the Volume Master File (VMF). Its EXPDT is set with $OPTIONS or the TAPE1 DD statement in UNLOAD30 steps. Files 2 through n are created with RETPD=0, bypassing the VMF update. This avoids cluttering the VMF with unnecessary data set name entries for the same volume.

6.2.4.1 Both TLMS and CA-ASM2 Control the Tapes

The advantages to using this method are:

- Full tape protection offered by TLMS.
- Provides full tape recycling capabilities (both master and duplex).
- CA-ASM2 tapes can be kept separately near the tape drive area (critical for online reloads).
- CA-ASM2 tapes can be new tapes. This reduces the chance of physical I/O errors.

The disadvantages to using this method are:

- Continual management and maintenance of the tape pools.
- If no tapes are available in the tape pool, a mount SCRTCH or PRIVAT is issued resulting in tape active but not in the tape pool.
- To solve the above problems, over allocation of tapes in the pool causes wasted tape media resources.
- Manual updates of the VMF for each volser in the tape pool. See Step 5 in the following procedure.

To use this method, follow this step-by-step procedure:

1. **Update $XTTMS in CAI.CAIMAC.**

   - First, indicate you want a TLMS interface by setting the variable &TAPEMGT accordingly:

     ```
     &TAPEMGT SETC  'TLMS2'  TLMS INTERFACE
     (for Ver. 5.2 and later)
     ```

     Note: If you have a TLMS version prior to 5.2, use 'TLMS'.

   - During the installation you chose a prefix for CA-ASM2 data set names. ASM2 was the default if you did not specify otherwise.

     To choose a different prefix for tape IDs, just replace ASM2 in the &PREFX statement with the prefix you want.

     ```
     &PREFX   SETC  'ASM2'    DSN PREFIX
     ```

     Note: Pick one naming scheme to use in $NTEXIT and use it consistently. Do not, for instance, change $NTEXIT with each version of CA-ASM2, making tape name release dependent.

   - Next, set the value for &TAPPOOL to 'YES':

     ```
     &TAPPOOL SETC  'YES'     CA-ASM2 TAPE POOL USED
     ```

   - Set the value of &DUPEDSN to 'YES' to insert the master tape's volser into duplex tape data set names, as follows:

     ```
     &DUPEDSN SETC 'YES' INSERT MASTERVOL INTO DUPE DSN
     ```

$XTTMS is now updated. To complete step 1, save $XTTMS into CAI.CAIMAC with the changes you made.

2. **Use the CA-ASM2 tape pool and CA-ASM2 exits to pass updates to the TLMS VMF. Verify or change certain values in $OPTIONS to facilitate this. Review $OPTIONS.**

   $TAPPOOL should be set to X'01'.
   $ARCHEXP should be set to 99365.
   $BKUPEXP should be set to 99365.
   $MNTOPT should be set to X'01'.
   $DUP#MAX should be specified.
   $DUPTMAX should be specified.

   The reason $MNTOPT should be set to X'01' is so CA-ASM2 automatically frees expired archive tapes. Backup tapes are automatically freed. For more details, see the explanation of $OPTIONS field $MNTOPT, located in the *CA-ASM2 Planning Guide*.

3. **Assemble the CA-ASM2 TLMS interface exits.**

   You can find sample JCL to do this in member ASMTLMS of SAMPJCL.

   - Verify the data set names of SYSLIB libraries specified in ASMTLMS.
   - In CAI.CAISRC, assemble member $FTEXIT and link it to CAI.CAILIB with the member name $FTEXIT. It is automatically assigned aliases of $FT and $FTTLMS2 in Version 5.2 and later.
   - In CAI.CAISRC, assemble member $NTEXIT and link it to CAI.CAILIB with the member name $NTEXIT. It is automatically assigned the aliases $NT and $NTTLMS2 (Version 5.2 and later).
   - In CAI.CAISRC, assemble member $TSINQ and link it to CAI.CAILIB with the member name $TSINQ. See $TSINQ installation requirements later in this chapter.

   You have now completed assembling.

4. **Ensure that the TLMS module (TLMSASM2) is in a library that is accessible by CA-ASM2 runs.**

5. **Have your TLMS systems programmer define the CA-ASM2 tapes to TLMS. Then initialize the physical tape volumes.**

   The tapes should have expiration dates of 99365, and data set names that begin with the prefix defined in $XTTMS.

6. **If you are vaulting the CA-ASM2 duplex tapes, make sure you set &DUPEDSN to YES.**

   Have your TLMS systems programmer define in the TLMS RMF data set a rule for each possible CA-ASM2 duplex tape, specifying the tape's full data set name. A rule specifying that one cycle is to be kept should be satisfactory in most cases. For example:

   ```
   TDAASM2.ARCH.DUPE/
   6DC    7HS
   ```

   See the TLMS User guide for more information.

   The same method applies for backup. If you are going to create and vault duplex tapes in the CA-ASM2 backup process, you must establish similar patterns for duplex backup volumes.

   You have now completed implementing the CA-ASM2 TLMS interface allowing both TLMS and CA-ASM2 to control the tapes and use the CA-ASM2 tape pool.

7. **Review the tape duplexing module.**

   Read $RSUTLMS in CAIMAC for information on maintaining duplex tapes automatically.

6.2.4.2 TLMS Controls the Tapes with CA-ASM2 Freeing the Tapes

This method provides the full tape protection offered by TLMS. In addition, the master tapes remain CA-ASM2 tapes (EXPDT=99365) until CA-ASM2 frees them without your manual intervention. The disadvantage of this method is that you cannot totally automate the release of duplex tapes from TLMS.

To use this method, follow this step-by-step procedure:

1. **Update $XTTMS in CAI.CAIMAC.**

   - First, indicate that you want a TLMS interface by setting the variable &TAPEMGT accordingly:

     ```
     &TAPEMGT SETC  'TLMS2'    TLMS INTERFACE
     ```

     (For Versions 5.2 and later)

     Note: If you have a TLMS version prior to 5.2, use 'TLMS'.

   - During the installation you chose a prefix for CA-ASM2 data set names. ASM2 was the default if you did not specify otherwise.

     To choose a different prefix for tape IDs, just replace ASM2 in the &PREFX statement with the prefix you want.

     ```
     &PREFX   SETC  'ASM2'    DSN PREFIX
     ```

     Note: Pick one naming scheme to use in $NTEXIT and use it consistently. Do not, for instance, change $NTEXIT with each version of CA-ASM2, making tape name release dependent.

   - Next, set the value for &TAPPOOL to 'NO':

     ```
     &TAPPOOL SETC  'NO'     CA-ASM2 TAPE POOL NOT USED
     ```

   - Place the volser of the CA-ASM2 master tape in the DSNAME of the duplex tapes created for it. This is essential since this method does not use the CA-ASM2 tape pool function. Set the value of &DUPEDSN to YES to insert the master tape's volser into duplex tape data set names. For example:

     ```
     &DUPEDSN SETC  'YES'   INSERT MASTERVOL INTO DUPE DSN
     ```

     $XTTMS has now been updated. To complete step 1, save $XTTMS into CAI.CAIMAC with the changes you made.

Caution

Use of the TLMS DBLTIME (Double OPEN) parameter can potentially cause a tape to be written over if it is opened twice in the same job as an output tape. Please review the DBLTIME parameter in the *CA-Dynam/TLMS Systems Programmer Guide*.

2. **Force CA-ASM2 to use TLMS scratch tape and CA-ASM2 exits to update the VMF as necessary.**

   - Verify or change certain values in $OPTIONS to facilitate this. Review $OPTIONS:

     $TAPPOOL should be set to X'00'.
     $ARCHEXP should be set to 99365.
     $BKUPEXP should be set to 99365.
     $MNTOPT should be set to X'01'.
     $DUP#MAX should be specified.
     $DUPTMAX should be specified.

     The reason $MNTOPT should be set to X'01' is so CA-ASM2 automatically frees expired master archive tapes. For more details, see the explanation of $OPTIONS field $MNTOPT, located in the *CA-ASM2 Planning Guide*.

3. **Assemble the CA-ASM2 TLMS interface exits.**

   You can find sample JCL to do this in member ASMTLMS of SAMPJCL.

   For the assembly step, you must specify three MACLIBs:

   CAI.CAIMAC to include TLMS and CA-ASM2 macros
   SYS1.MACLIB to include IBM macros
   SYS1.AMODGEN to include IBM macros

   These may not be their true data set names in your installation. Check with the person responsible for TLMS to verify in which MACLIB the TLMS macros were installed.

   - In CAI.CAISRC, assemble member $FTEXIT and link it to CAI.CAILIB with the member name $FTEXIT. It is automatically assigned aliases of $FT and $FTTLMS2.

   - In CAI.CAISRC, assemble member $NTEXIT and link it to CAI.CAILIB with the member name $NTEXIT. It is automatically assigned the aliases $NT and $NTTLMS2 (Versions 5.2 and later).

   - In CAI.CAISRC, assemble member $TSINQ and link it to CAI.CAILIB with the member name $TSINQ. See $TSINQ installation requirements later in this chapter.

     You have now completed assembling.

4. **Ensure that the TLMS module (TLMSASM2) is in a library that is accessible by CA-ASM2 runs.**

5. **Change the CA-ASM2 JCL procedures.**

   - Change the CA-ASM2 archive PROCs to handle the process for expiring duplex tapes. In each PROC containing a duplex step (TPBKUP70 in ASM2SYSA, ASM2EXPA, and ASM2EXPP), modify the OUTAPE DD statement to include EXPDT=99001 in the LABEL parameter. This causes TLMS to keep the most recent duplex tape of any master tape. Verify that this is the case for TLMS.

- If you plan to duplex backup tapes and archive tapes, change the CA-ASM2 backup procedures to handle the process of expiring duplex tapes. In each procedure containing the duplex step (TPBKUP70 in ASM2SYSB and ASM2EXPB), you can either (1) set up a manual procedure and use EXPDT=99001 for OUTTAPE in the TPBKUP70 step, or (2) assign duplex tapes a retention period equal to the maximum retention for a backed up data set. The drawback of using the retention period is that most duplex tapes are kept longer than necessary.

  CA-ASM2 backup master tapes are automatically purged from CA-ASM2 and TLMS. You receive a message that the tape is purged. When you receive the message, have your tape librarian ensure that the associated duplex tape has expired.

6. **If you are vaulting the CA-ASM2 duplex tapes, make sure you set &DUPEDSN to YES.**

   Have your TLMS systems programmer define in the TLMS RMF data set a rule for each possible CA-ASM2 duplex tape, specifying the tape's full data set name. A rule specifying that one cycle is to be kept should be satisfactory in most cases. Example:

   ```
   TDAASM2.ARCH.DUPE/
   6DC    7HS
   ```

   See the TLMS User guide for more information.

   The same method applies for backup. If you are going to create and vault duplex tapes in the CA-ASM2 backup process, you must establish similar patterns for duplex backup volumes.

   You have now completed implementing the CA-ASM2 TLMS interface giving TLMS control of the tapes with CA-ASM2 controlling tape retention.

7. **Review the tape duplexing module.**

   Read $RSUTLMS in CAIMAC for information on maintaining duplex tapes automatically.

6.2.4.3 $NTEXIT/$FTEXIT Interface

Review the following points if a CA-ASM2 TLMS interface is written in $NTEXIT or $FTEXIT:

1. Check the sample code that is generated for TLMS in members $NTEXIT and $FTEXIT of CAI.CAISRC.
2. Check the TLMS documentation for information on the TLMSASM2 module used to interface between CA-ASM2 and TLMS.
3. Add two DD statements to every CA-ASM2 step (archive, backup, $MAINT, forward merge, $DEFRAG or copy) that can call for a new output tape or free a CA-ASM2 tape. The DD statements for Version 5.2 and later should look like this:

```
//CAIVMFI DD DISP=SHR,DSN=your.master.file
//CAIVMF  DD DISP=SHR,DSN=your.master.file
```

4. $TAPEMGT must be set to TLMS2.

6.2.4.4 CA-ASM2 Controls Tapes

This method uses only the CA-ASM2 tape pool to control tapes and Computer Associates definitely does **not** recommend it. If you use it, you lose the tape protection offered by TLMS.

To use this method, follow this procedure:

1. Verify or change certain values in $OPTIONS to use CA-ASM2 tapes in the TLMS environment. Review $OPTIONS:

   $TAPPOOL should be set to X'01'.
   $ARCHEXP should be set to 98000.
   $BKUPEXP should be set to 98000.
   $MNTOPT should be set to X'01'.
   $DUP#MAX should be specified.
   $DUPTMAX should be specified.

   The reason $MNTOPT should be set to X'01' is so CA-ASM2 automatically frees expired archive tapes.

2. Ensure that the volsers assigned to CA-ASM2 tapes in the tape pools do not match any volsers defined to TLMS.

6.2.5 $TSINQ Installation Requirements for TLMS Sites

$TSINQ is used by the maintenance utility, $MAINT, to determine when to expire full-volume dump tapes created by $DEFRAG. If your installation does not run the $DEFRAG utility and has no plans to, you may skip linking the VMF I/O module TLMSVMRW. The following JCL should be run to create the I/O module TLMSVMRW:

```
//STEP1    EXEC PGM=HEWL,PARM='LIST,XREF'
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(2,2))
//TLMSLIB  DD  DSN=&tlmspfx..CAILIB,DISP=SHR
//SYSLMOD  DD  DSN=&prefix..CAI.CAILIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSLIN   DD  *
 REPLACE TLMSXTRS
 REPLACE TLMSUNCT
 INCLUDE TLMSLIB(TLMSXTRS)
 NAME    TLMSVMRW(R)
/*
//
```

After installing $TSINQ and TLMSVMRW in CAI.CAILIB, a DD statement defining the VMF must be added to the JCL for $MAINT:

```
//CAIVMFI  DD  DSN=&tlmspfx..VMF,DISP=SHR
```

No updates are made to the VMF.

6.3 Security Systems Support

CA-ASM2 provides support for the following commercially available data security systems:

- CA-ACF2
- CA-Top Secret
- RACF

Access to the Extended ISPF User Interface is controlled by external security having 6 different authority levels. See the appendix in the *CA-ASM2 ISPF User Interface Guide* relating to "Extended ISPF Applications" for further details.

- PROG authority
  - Can only access data sets beginning with their userid
  - Can access the masks defined in the exception list in L1AAXMOD
  - All restore requests are queued (deferred execution)
  - All update commands are executed online
  - NEWVOL is not allwed
  - No mass commands can be used
  - No application definitions
  - No VSAM exception definitions
  - No full or incremental restore
  - No IPC jobs can be performed
  - Only options 0, 1, 2, and F are on the main menu
- PRG1 authority
  - Same rules as PROG, except that requests are executed in batch mode
  - Depending on the parameter X2SUB in customization program L1AAXMOD, you receive the JCL in Edit, or the JCL is submitted automatically.
- PRG2 authority
  - Same rules as PROG, except that the user can access all data sets
  - Provides queued reloads
- PRG3 authority
  - All files can be accessed
  - Only options 0, 1, 2, and F are on the main menu
  - All commands and options are permitted on these restricted menu options
  - Provides batch reloads
- SYST authority

  - Can perform all operations
  - No panel with the needed tapes are shown
- OPER authority
  - Same as SYST user, except that just before the Edit of the JCL, a panel displays all needed tapes for the restore operation

In L1AAXMOD, the parameter X2DSNSEC can be activated to check if the user is allowed to READ or UPDATE the requested data set. This extra check can be deactivated for SYST and OPER users.

These authority levels are translated to a CACMD level.

Authority	CACMD level
SYST	L1RALL
OPER	L1BALLT
PRG3	L1BATAL
PRG2	L1RQUEAL
PRG1	L1RBATON
PROG	none (access by default)

6.3.1 CA-ACF2

CA-ACF2 support is activated through the $OPTIONS field $SOFTWAR. The option indicating that the site has CA-ACF2 installed and would like to perform security validation through $AUTHEXT should be set to 1 (the X'40' bit). If you did not set this option to 1 when initially running the $OPTIONS task L140IH, you should do so now and generate and submit the task again to reassemble $OPTIONS.

CA-ACF2 support is provided by modules $AUTHEXT and $AUTHXT3. Source for these modules is given in CAI.CAISRC. $AUTHEXT is distributed in load module form in CAI.CAILIB. You are responsible for assembling and link editing $AUTHXT3. To accomplish this, in the assembly step concatenate the CA-ASM2 CAI.CAIMAC data set and the macro library containing CA-ACF2 macros to your standard SYSLIBs.

The sample JCL shown next is located in member ASMACF2 of SAMPJCL. Be sure to replace the high-level index in the JCL, shown as DSN=prefix with the CA-ASM2 prefix. In the link-edit step, use the CA-ACF2 library containing the CA-ACF2 load modules as a SYSLIB.

```
//ASMACF2 JOB (9000,9000),PRGMMR,CLASS=K
/*JOBPARM R=XXXX
//******************************************************************
//*
//* CHANGE ALL OCCURRENCES OF PREFIX TO YOUR ASM2 PREFIX
//* CHANGE ACF2.ACFMAC TO THE CORRECT NAME FOR ACF2 MACRO LIBRARY
//* CHANGE ACF2.ACFAMOD TO THE CORRECT NAME FOR ACF2 DIST LIBRARY
//*
//******************************************************************
//JS001  EXEC  ASMHCL,PARM.ASM='OBJ,NODECK',COND=(0,NE),
//             PARM.LKED='MAP,LET,LIST'
//ASM.SYSLIB   DD DSN=PREFIX.CAIMAC,DISP=SHR
//             DD DSN=SYS1.MACLIB,DISP=SHR
//             DD DSN=SYS1.AMODGEN,DISP=SHR
//             DD DSN=ACF2.ACFMAC,DISP=SHR
//ASM.SYSIN    DD DSN=PREFIX.CAISRC($AUTHXT3),DISP=SHR
//LKED.SYSLMOD DD DSN=PREFIX.CAILIB,DISP=SHR
//LKED.SYSIN   DD *
  NAME $AUTHXT3(R)
//LKED.SYSLIB  DD DSN=ACF2.ACFAMOD,DISP=SHR
```

Note: $AUTHXT3 must not be link edited as reentrant (do not use the LINKEDIT parameter RENT or REFR).

6.3.1.1 CA-ASM2 and CA-ACF2 Considerations

Computer Associates suggests that you coordinate the following considerations with the person at your installation responsible for the maintenance of CA-ACF2.

Define a special logon ID for normal CA-ASM2 batch processing. Define this logon ID as NON-CNCL, RESTRICT, and SUBAUTH to limit its use to production CA-ASM2 jobs.

Add the following to the ACFFDR to control use of CA-ASM2 programs:

```
SET CONTROL(GSO)
INSERT PPGM PGMMASK($COPYTP,$TPCOPY,$FM,$DEFRAG)
```

to limit use of these programs to the CA-ASM2 LOGONID. In addition, define some CA-ASM2 programs as maintenance programs to avoid excessive logging by CA-ACF2 to SMF.

```
SET CONTROL(GSO)
INSERT MAINT.ASM2 LID(logonid),LIB('SYS1,LINKLIB'),
                  PGM=($DASDMNT,$MIGRATE,$MAINT)
```

Note: After the PPGM and MAINT records have been inserted, issue the following commands to activate the records:

```
F ACF2,REFRESH(PPGM)
F ACF2,REFRESH(MAINT)
```

To protect the CA-ASM2 system, add other CA-ACF2 rules. The best approach is to use program pathing to control the program's access to the IPC and journal data sets rather than the individual users. There are examples of rule sets for you to follow located in member ACF2RULE in SAMPJCL.

Browsing and making maintenance changes to the CA-ASM2 IPC using CA-ASM2 ISPF support is best controlled by giving read and write authority of the IPC and Journal clusters to authorized data administrators.

6.3.1.2 ISPF Interface Security

The IPC maintenance function of the ISPF interface (option 8) requires a rule to be added with the following format to allow you to use this option.

```
$KEY(ASM2CU10) TYPE(PAN) UID(XXXX) ALLOW
```

To control access to the Extended ISPF functions (option 7):

- Define the CAC(= CACMD) resource type, if not done previously.

```
SET CONTROL(GSO)
CHANGE RESDIR TYPES(D-CAC)
```

- You must then use the refresh command for resident rules.

```
F ACF2,REFRESH(RESDIR)
```

- Create a new member in the CA-ACF2 rules PDS.

```
$KEY(.......) TYPE(CAC)
UID(uidstring/mask) ALLOW
```

- The KEY in the previous rule is a CACMD level. For example, specify $KEY(L1RALL) for system programmers.
- Repeat the previous lines for all needed access levels.
- Enter ACF mode by typing the TSO command ACF.
- Set the appropriate CA-ACF2 resource for CA-ASM2 levels by entering SET RES(CAC).
- Compile the newly created rules by entering COMPILE pdsname(memname).
- Update CA-ACF2 with the new resource rule by entering STORE*.
- End CA-ACF2 native mode by entering END.
- If you use resident resource rule sets, you must rebuild the resident rule set directories as shown next.

```
F ACF2,REBUILD(CAC)
```

The levels are described on page 6-23.

6.3.1.3 Access Authority

CA-ASM2 issues a call to security through exits. If an RC=0 is passed in R(15), access is allowed. The check is based on DSN checking by the USERID set up in the security rules. A return code of other than zero passed back from the security check disallows access to information. The level of data set access authority required for each of the applicable CA-ASM2 commands is shown next.

Command	Authority Required
$AI	None
$AR	ALLOC
$BI	None
$BK	READ
$CI	None
$CU	ALLOC
$DA	ALLOC
$DB	ALLOC
$MIGRATE	ALLOC
$RA	ALLOC
$RB	ALLOC
$SM	WRITE
$UA	ALLOC
$UB	ALLOC
$US	None

6.3.2 CA-Top Secret

If you have installed CA-ASM2 on a system with CA-Top Secret, you can activate the CA-Top Secret interface using the $SOFTWAR field of $OPTIONS. Bit X'04' is used to request either RACF or CA-Top Secret support. CA-Top Secret users must set this bit ON to activate the CA-Top Secret interface.

If you have Always-Call support, you can activate this feature using the $RACFOPT field of $OPTIONS. When the Always-Call feature is activated, CA-ASM2 always calls the CA-Top Secret system before accessing a data set even if the data set is not security indicated.

CA-Top Secret support is accomplished by the following exits which are assembled and linked by the base installation.

$AUTHEXT
$AUTHXT2
$AUTHXT4
$ARIROUT

All of the above modules are linked to CAI.CAILIB by the SMP install, however the source to these routines is also provided in CAI.CAISRC to allow your installation to include additional validation if needed. The source to $ARIROUT is distributed in member ARIROUT of CAI.CAISRC. $ARIROUT is an alias assigned when the module is linked. Member ASMRACF of SAMPJCL has the JCL necessary to assemble these modules if you choose to include further security processing. Remove the execute statement to assemble M2HEINTY from ASMRACF (this is for RACF installations only). If module $ARIROUT is changed, you must also run job LKEDRACF in SAMPJCL.

Browsing and making maintenance changes to the CA-ASM2 IPC using CA-ASM2 ISPF support is best controlled by giving read and write authority of the IPC and Journal clusters to authorized data administrators.

6.3.2.1 ISPF Interface Security

The CA-Top Secret statements required to allow you access to the IPC Maintenance option of the CA-ASM2 ISPF interface (option 8) has the following format.

```
$TSS ADD (OWNER) PANEL(ASM2CU10)
$TSS PERMIT(XXXX) PANEL(ASM2CU10) ACCESS(READ)
```

To allow access to the Extended ISPF functions (option 7), enter the following statements:

- Protect all access levels by:

  TSS ADDTO(deptname)
  CACMD(one or more of the levels
  L1RALL,L1BALLT,L1BATAL,L1RQUEAL,L1RBATON)

- Allow users to specific levels by:

```
TSS PERMIT(userid)CACMD(one of the levels)
```

The CACMD parameters can be any of those described on page 6-26.

6.3.2.2 Access Authority

CA-ASM2 issues a call to security through exits. If an RC=0 is passed in R(15), access is allowed. The check is based on DSN checking by the USERID set up in the security rules. A return code of other than zero passed back from the security check disallows access to information. The level of data set access authority required for each of the applicable commands is shown next.

Command	Authority Required
$AI	None
$AR	CREATE and SCRATCH
$BI	None
$BK	READ
$CI	None
$CU	CREATE and SCRATCH
$DA	CREATE and SCRATCH
$DB	CREATE and SCRATCH
$MIGRATE	CREATE and SCRATCH
$RA	CREATE and SCRATCH
$RB	CREATE and SCRATCH
$SM	WRITE
$UA	CREATE and SCRATCH
$UB	CREATE and SCRATCH
$US	READ

6.3.3 RACF

If you have installed CA-ASM2 on a system with RACF, you can activate the RACF interface using the $SOFTWAR field of $OPTIONS (bit X'04' which is also used to activate CA-Top Secret).

If you have Always-Call support, you can activate this feature using the $RACFOPT field of $OPTIONS. When the Always-Call feature is activated, CA-ASM2 always calls the RACF system before accessing a data set even if the data set is not RACF indicated.

RACF support is accomplished by the following exits:

$AUTHEXT
$AUTHXT2
$AUTHXT4
M2HEINTY
$ARIROUT

Source modules are in CAI.CAISRC. Load modules are in CAI.CAILIB, except for M2HEINTY, which must be assembled and link edited. See member ASMRACF of SAMPJCL for sample JCL to assemble and link M2HEINTY.

Module M2HEINTY is required for RACF support and must be assembled during CA-ASM2 installation. The remaining modules are linked to CAI.CAILIB by the

SMP install. The source to these routines is also provided in CAI.CAISRC to allow your installation to include additional validation if needed. The source to $ARIROUT is distributed in member ARIROUT of CAI.CAISRC. $ARIROUT is an alias assigned when the module is linked. Member ASMRACF of SAMPJCL has the JCL necessary to assemble these modules if you choose to include further security processing.

6.3.3.1 ISPF Interface Security

The IPC maintenance option (option 8) requires the inclusion of the following entries to be added to the RACF system.

A new class descriptor table entry,

```
ICHERCDE CLASS=PA@EL,
   ID=nnn,
   FIRST=ALPHA,
   OTHER=ALPHANUM
   POSIT=nn,
   DFTUACC=NONE
```

the following router table entry,

```
ICHRFRTB CLASS=PA@EL,
   ACTION=RACF
```

and the following RACF REDEFINE command:

```
RDEFINE PA@EL (ASM2CU10)
   UACC(READ)
```

To control access to the Extended ISPF functions (option 7):

- Edit SYS1.PARMLIB member IKJTSO00 and add programs L1AAX051, L1AAX069, L1AAX075 and L1AAX105 to labels AUTHCMD and AUTHPGM.
- Issue the TSO command PARMLIB UPDATE(00) to refresh these updates.
- Verify if another Computer Associates product had the CA@MD requirement. If not, execute the following steps.
- Add CA@MD in the Class Descriptor table.

```
ICHERCDE CLASS=CA@MD,                   x
      ID=128,                           x
      FIRST=ALPHA,                      x
      OTHER=ANY,                        x
      POSIT=25,                         x
      DFTUACC=NONE
```

- Add CA@MD in the RACF Router table.

```
ICHRFRTB CLASS=CA@MD,                   x
      ACTION=RACF
```

- IPL your system for this last update
- Activate CA@MD via: `SETR CLASSACT(CA@MD)`
- Protect the 5 levels of this interface by:

```
RDEFINE CA@MD(cacmd name) UACC(NONE)
```

Repeat this command for all five levels

- Permit the users to one of these levels by:

```
PERMIT cacmdname CLASS(CA@MD) ID(user1,user2)ACCESS(READ)
```

The levels are described on 6-23.

6.3.3.2 RACF Profiles

CA-ASM2 builds a special RACF profile to cover each data set in the archive or backup catalog that had a discrete profile. If two or more unloaded data sets with discrete profiles have the same name, one RACF profile covers them all. That profile is modeled after the discrete profile of the data set most recently unloaded (whether in the archive or the backup catalog).

If a data set is covered by a generic profile and does not have a discrete profile of its own, CA-ASM2 turns off a flag (X'80' bit in field PXSEC) in the catalog record(s) for that data set, and does not build a RACF profile.

The name of the RACF profile is the name of the data set, with all levels of the name below the highest changed to lowercase. For example, the data set name USR013.PROJECT5.ASM generates a profile named USR013.project5.asm. The profile may be either a multi- or a single-volume profile, depending on whether the data set occurs in both the archive catalog and the backup catalog, or only in one of them. The volser(s) associated with a RACF profile are ASMBKP for backed up data sets, and ASMARC for archived data sets.

The RACF database increases in size. The maximum possible size is double the current size. While the data set still exists on disk and is being unloaded, CA-ASM2 creates a second profile (if it does not already exist). This second profile is the RACF profile with volser ASMBKP or ASMARC. (Once the data set is archived, it is deleted from disk with its original profile.)

It is possible to change the CA-ASM2 volser(s) in the profile ASMBKP and ASMARC to something else if your shop has some reason for doing this, for example, if you run multiple versions of CA-ASM2 (controlled by $OPTIONS) and there is the possibility that a protected data set that exists in the catalog(s) of one version of CA-ASM2 could also exist in the catalogs of another. This is important because of the method used to delete the RACF profiles. When there is no entry in the CA-ASM2 catalog **being processed**, the RACF profile is deleted. So, if the data set exists in two CA-ASM2 environments, the profile is deleted prematurely and errors result in referencing that data set in the other environment.

When a data set is backed up and a RACF profile exists for it (because it was previously backed up), the old RACF profile is deleted and a new one is added. In this way, CA-ASM2 avoids saving a new profile for each copy of the data set but still maintains the latest possible attributes (such as the access list).

When CA-ASM2 reloads a data set that has a RACF profile, CA-ASM2 knows it needs a discrete profile online. In this case, CA-ASM2 performs the following four steps to ensure that it has a discrete profile:

1. CA-ASM2 checks to see whether the RACF system has a discrete profile for the data set name associated with the target disk volume. If such a profile exists, CA-ASM2 does nothing further.
2. If the RACF system does not have a discrete profile for the name associated with the volume, CA-ASM2 looks for a profile associated with the home volume. (The home volume is the one from which the data set was unloaded.) If the RACF system has a discrete profile for the data set name associated with the home volume, CA-ASM2 uses this profile to model a profile for the reloaded data set.
3. If there is no discrete profile for the data set name associated with either volume, CA-ASM2 uses the lowercase profile created at unload as a model to define a new discrete profile for the data set.
4. If for some reason CA-ASM2 can neither find nor create a discrete profile for the reloaded data set, CA-ASM2 turns on the RACF bit in the DSCB and issues a warning message that a RACF error occurred. It is then up to the user to issue the commands to define a discrete profile. This probably requires someone in the security group with operations authority to issue an ADDSD command with NOSET.

This procedure maintains the most recent attributes available.

6.3.3.3 Miscellaneous CA-ASM2 RACF Considerations

When a data set (with a discrete profile) is migrated from disk-to-disk by $MIGRATE, a new discrete profile is built for the output disk volume. This allows $MIGRATE (with PARM=NOSCRATCH) to create a copy without affecting the profile for the original volume. However, this could be a problem if your shop does not allow multiple RACF profiles for the same data set name. You can alter this processing in the $ARIROUT module.

When CA-ASM2 creates a RACF profile for a backed up or archived data set, the disk profile is not changed but is deleted if the disk data set is deleted. All RACF commands can still function on the disk profile as they do when CA-ASM2 is not in the system.

When you issue a $AR or $BK command to unload a data set, CA-ASM2 performs a RACHECK to look for a discrete or generic profile that covers the data set. When you issue a $RA, $RB, $UA, $UB, $DA, or $DB command, CA-ASM2 performs a RACHECK for either a RACF profile or a generic profile. In this case, the RACHECK finds the following: (1) a RACF profile if the data set had a discrete profile at unload time, (2) a generic profile if the data set had only a generic profile, or (3) nothing if the data set had neither.

There is a subtle problem that may arise with RACF profiles. If a user is allowed access at the time a data set is unloaded, the command is allowed even though access has since been denied to the user for the disk data set. Also, if the user was not allowed access at the time of the most recent unload, the command is denied even if the user has since been allowed access to the disk data set.

Expanding the security database by adding RACF profiles may cause problems in your installation. The solution is to modify ARIROUT to prefix the data set names used in the RACF profiles and segregate that prefix to a unique RACF database. On any profile built by CA-ASM2 for either reload or migration, the defining group of the new profile is that of the user actually moving the data. In other words, queued reload propagates the group of the user ID specified on the queued reload job.

If your installation allows data sets with the same name to exist on different disk volumes and have different discrete profiles, a problem occurs if CA-ASM2 unloads two or more of the data sets. Only one of the CA-ASM2 RACF profiles that CA-ASM2 creates can be valid. To solve this problem, modify ARIROUT to make the data set name or volser used on the RACF profile unique.

A RACF command does not work on a CA-ASM2 RACF profile because the profile contains both uppercase and lowercase characters. But the command, including the data set name, is always in uppercase (it is translated to uppercase during processing).

If your installation has RACF exits and you want to use the installation parameter list (keyword INSTLN on the RACHECK and RACDEF macros) or the application name (keyword APPL on the RACHECK macro), you can modify the CA-ASM2 module ARIROUT to build and pass the parameter list and/or application name. If ARIROUT does build the installation parameter list, it is passed to both RACHECK and RACDEF.

If a RACHECK exit exists that allows access to a data set without completing the RACHECK logic, errors can occur. These errors are usually S0C4 abends. It is the responsibility of the RACHECK exit to check the RACHECK parameter list of the calling program for a CSA call (X'01' bit set at offset +4 indicates a CSA call), and then, to GETMAIN an area and build a dummy profile.

Browsing and making maintenance changes to the CA-ASM2 IPC using CA-ASM2 ISPF support is best controlled by giving read and write authority of the IPC and Journal clusters to authorized data administrators.

6.3.3.4 Access Authority

CA-ASM2 issues a call to security through exits. If an RC=0 is passed in R(15), access is allowed. The check is based on DSN checking by the USERID set up in the security rules. A return code of other than zero passed back from the security check disallows access to information. The level of data set access authority required for each of the applicable CA-ASM2 commands is shown next.

Command	Authority Required
$AI	None
$AR	ALTER
$BI	None

$BK	READ
$CI	None
$CU	ALTER
$DA	ALTER
$DB	ALTER
$MIGRATE	ALTER
$RA	READ
$RB	READ
$SM	ALTER
$UA	ALTER
$UB	ALTER
$US	READ

6.3.3.5 Override ADSP Option

When the ADSP (Automatic Data Set Protection) option is used, any permanent data set that is created automatically receives a discrete profile. In shops that require the use of the ADSP option, CA-ASM2 can optionally override the security system's discrete profile built when it allocates a data set for reload.

When the CA-ASM2 override ADSP option is used, CA-ASM2 determines if reloading user ID has the ADSP option on. If the ADSP option is on, CA-ASM2 checks the existing profile's create date to ensure the profile was just created. Since there is no sure way to determine that the existing profile was created by the ADSP option, CA-ASM2 assumes the profile was built by ADSP if the profile's create date is equal to the current date, or if the profile's create date is equal to yesterday's date and the current time is before 1:00 AM. If CA-ASM2 determines the existing profile was built by ADSP, CA-ASM2 deletes the existing profile and uses the RACF profile as a model to build a new profile (see 6.3.3.2, “RACF Profiles” on page 6-33. Any existing profile found not to be created by ADSP is kept to cover the reloaded data set.

To enable the CA-ASM2 override ADSP option, indicate ADSP for the $RACFOPT field of $OPTIONS.

6.4 ISPF Support

CA-ASM2 provides support from ISPF menus for the $AR, $BK, $AI, $BI, $DA, $DB, $UA, $UB, $QM, $SM, $US, $RA, and $RB commands. Only the queued reload forms of $RA and $RB are available under ISPF. $RSVP commands are also available from SPF menus.

The CA-ASM2 dialog manager (ASM2SPF), provides support for ISPF. This dialog function uses ISPF Panels, Messages and the ISPF Dialog Management Services to invoke CA-ASM2 online commands. The standard SPF support provided with CA-ASM2 is for the Interactive System Productivity Facility.

TSO/E users can execute all CA-ASM2 commands using the ISPF interface. Non-TSO/E users are limited to executing CA-ASM2 commands which do not require APF authorization.

1. ISPF support is facilitated by the following four files from the CA-ASM2 installation tape:

   CAI.CAICLIB
   CAI.CAIISPP
   CAI.CAIISPM
   CAI.ASM2V.ARCH.RSVP.TABLE

   These files contain the ISPF clists, panels and messages, and the VSAM CA-RSVP Saved Commands Table, respectively. The Saved Commands Table saves $RSVP commands built in ISPF Mode (using the CA-RSVP panels) for later modification and reuse. These four files were down loaded earlier in Step 3 of the installation.

   Concatenate the clist, panel and message libraries (CAI.CAICLIB, CAI.CAIISPP and CAI.CAIISPM) to the users' ISPLLIB, ISPPLIB and ISPMLIB allocations. To use an alternative method, copy the members of these three data sets to the appropriate ISPF libraries. For Extended functions, add CAICLIB to the SYSPROC concatenation.

2. Edit CAI.CAIISPP member ASM2RS00 using ISPF EDIT. Change the default setting of variable &ASRSDFLT in the INIT section of the panel to the name selected for the VSAM CA-RSVP table CAI.ASM2V.ARCH.RSVP.TABLE (must be in uppercase), to whatever you renamed the VSAM CA-RSVP data set. If this is not done or if done incorrectly, it results in "INIT FAILED" messages.

3. Make sure the load module ASM2SPF is in either a linklist library, a STEPLIB library, or a library allocated to ISPLLIB or ISPLOAD.

4. Test the CA-ASM2 dialog using the ISPF DIALOG TEST option. Select the DIALOG TEST option on the Primary Option Menu. Then, select the FUNCTIONS option on the DIALOG TEST OPTION MENU. Enter the PROGRAM name ASM2SPF on the INVOKE DIALOG FUNCTION MENU in the PGM field. Enter RSVP on the PARM field. This should cause the CA-ASM2 Primary Selection panel to display. You can then test the CA-ASM2 functions by entering the appropriate selection codes.

5. Update your Primary Option Menu to include the CA-ASM2 select code for the ASM2SPF program when you are satisfied with the dialog's operation. A sample primary menu is provided in SAMPJCL with the name VISR@PRI. You can use it as is or make the following changes to your installation's ISR@PRIM menu.

   In the)BODY section of the panel, add the following display line:

   ```
   %  A +ASM2 - The Automated Space Management System
   ```

   In the)PROC section of the panel, add the following option line to the &ZSEL set command:

   ```
   A,'PGM(ASM2SPF) NOCHECK' /* CA-ASM2 ISPF */
   ```

   This causes the CA-ASM2 dialog manager to be selected when you enter option A.

6. TSO/E environments should utilize TSOEXEC services for invoking CA-ASM2 commands under ISPF. See $MISCOPT field in $OPTIONS.

7. Refer to the associated tutorial panels for instructions to use the CA-ASM2 ISPF panels. You may wish to print copies of the tutorials and distribute them to your users to let them know about these new facilities.

Note: If the CAI.CAILIB library is allocated to ISPLLIB only, all CA-ASM2 commands cannot be executed by CA-ASM2 ISPF dialog support. All CA-ASM2 commands are executed as subtasks by the CA-ASM2 ISPF dialog support and the CAI.CAILIB library must be allocated to //STEPLIB or be a linklist library.

6.5 VSAM List Utility

The $VLIST utility is a batch program that lists VSAM catalogs. If you run $VLIST, it must run as an authorized program, or you must supply the catalog master password if the catalog contains any password-protected data sets.

The following options are provided:

1. Bypass Password Checking: This is the default mode of operation. The $VLIST program is link edited as authorized and is run from an authorized library. In this mode, if a user does not supply the master password for a catalog, the catalog is listed anyway, bypassing password checking. No password information is reported so there is no exposure to unauthorized data set access. If a user supplies a password in the LISTCAT command, it is checked and the request is rejected if an incorrect password is provided.

2. Explicit Password Checking: If for some reason, you want to allow some or all catalogs to be listed only with specification of the proper master password, you should relink $VLIST as unauthorized. In this mode, catalogs containing password-protected data sets can only be listed if a user supplies the proper catalog master password. The most common way to supply the password is on the LISTCAT statement. The password is **not** printed.

3. Implicit Password Checking: If you are running $VLIST unauthorized but want some catalogs to be listed only with specification of the master password on the LISTCAT command, you must specify the catalog and the master password in a table that $VLIST uses if a password is not entered on the LISTCAT command. Generate this table by updating and assembling member $VLIST$ from CAI.CAISRC.

   Identify catalogs and their passwords by coding in $VLIST$ a macro for each catalog like the following example:

   ```
   CAT  CATALOG.VCATABC,MSTRPSWD
   ```

   See the *CA-ASM2 System Reference Guide* for details on how to run $VLIST.

6.6 Building the Stand-Alone Restore (SAR) Utility

During the base product installation, JCL PROCs called ASM2SART (for the tape version of SAR) and ASMSARD (for the disk version) were generated. When you run ASM2SART or ASMSARD, CA-ASM2 builds the SAR IPL utility on disk or tape in the following steps:

1. Link edits all the SAR load modules to build a module called $SAR.

2. Initializes a scratch tape (tape only).

3. When running the ASM2SART to create the stand-alone tape (SAR), remember to change BLP to No Label (NL) or Standard Label (SL) in the GENSAR step. Failure to do so causes a 'Disable Wait State' at stand-alone IPL time.

4. Executes the GENSAR step to run the $SABLD program and put the SAR utility on tape or disk.

   Before running ASM2SARD, edit the JCL to select the DASD volume you want. You can also change the data set name from the one that ASM2SART or ASM2SARD selects by default.

   Edit SAMPJCL member SARCONSL and update the following:

   - Console address statements defining the consoles used to run SAR in the format:

     ```
     CONSOLE(cuu,devtype)
     ```

     where cuu is the device address and devtype is the device type: 1052 - printer console and 3270 - display console. You can use any locally attached 3270 as a console, and define one master and up to 15 alternate consoles. SAR does not support integrated consoles on 370/168 CPUs.

     When ASM2SART or ASM2SARD is run, the output includes a table of the consoles defined in the GENSAR step, and a message giving the relative core location of the table:

     ```
     $SA0003I  offset IS OFFSET FOR CONSOLE TABLE
     ```

     Record this offset, so that someone who needs to run SAR from a new console can change the console address for the duration of that run (see "Stand-Alone Restore" in the *CA-ASM2 System Reference Guide)*.

     To change the table permanently you must build a new SAR. When you come to Step 3, specify all the consoles you want available for SAR, both old and new.

- (optional - disk only) The statement FORCEIPL tells $SABLD that it can write the bootstrap portion of SAR over any existing IPL text. Without this parameter, ASM2SARD fails if $SABLD finds IPL text already on track 0, cylinder 0 of the chosen volume.

Chapter 7. Installing Cumulative Maintenance

A cumulative maintenance tape is distributed periodically on a standard labeled, 6250 BPI tape reel (3420) or 38K BPI tape cartridge (3480), which can be processed by SMP/E. This maintenance tape contains all published official PTF SYSMODs for the current version of CA-ASM2.

A partitioned data set containing all the sample JCL is provided on the tape. All JCL needed to perform maintenance is found in the ninth data set and is in IEBCOPY unload format.

To load the maintenance sample JCL, use the sample provided next. Modify the tape volume serial number to the current volume serial number of the cumulative maintenance.

The volume serial number for cumulative maintenance tapes follows the format ***pcyymm***, where *pc* is the product code and *yymm* is the year and month of the tape. Please see the external label of the tape for the current volume serial number. To upgrade your product to the current maintenance level, complete the instructions that follow.

1. Check contents of this package against the enclosed packing list. Call your Technical Support organization for CA-ASM2 if you encounter any discrepancies.

2. Review any enclosed PIPs and PTFs to determine whether they pertain to your environment.

3. Download SAMPJCL from the maintenance tape using the following JCL:

```
//jobname  JOB (acct info),CLASS=a,MSGCLASS=x
//IEBCOPY  EXEC PGM=IEBCOPY,REGION=1024K
//SYSPRINT  DD SYSOUT=*
//INTAPE    DD DSN=CAI.SAMPJCL,
//             DISP=OLD,UNIT=tape,
//             VOL=SER=L1nnnn,
//             LABEL=(9,SL,EXPDT=98000)
//OUTDISK   DD DSN=your.prefix.SAMPJCL,DISP=SHR
//SYSUT3    DD UNIT=SYSDA,SPACE=(TRK,(5,1))
//SYSUT4    DD UNIT=SYSDA,SPACE=(TRK,(5,1))
//SYSIN     DD *
COPY INDD=((INTAPE,R)),OUTDD=OUTDISK
/*
```

Caution

Since combining the Installation and Maintenance tasks on one tape, the SAMPJCL and PPOPTION data sets are now complete replacement data sets. If you have altered any of the current members and wish to save them, either load the data sets to new data set names or rename the members you have changed.

CAIMAC and CAISRC members may be replaced when applying maintenance.

If you have altered members in these data sets you may wish to save them in another data set or rename them prior to applying maintenance.

4. Select SAMPJCL member MNTNOSMP to copy down changes to PPOPTION. Use the L1CHANGE ISPF edit macro from the install to change job control statement, procedure, and zone names.

5. Select job MNT1REC, the SMP/E RECEIVE job. Use the L1CHANGE ISPF edit macro from the install to change job control statement, procedure and zone names. Edit the MNT1REC job and make the necessary changes to the UNIT=tape and prefix.SAMPJCL lines.

   The following members of SAMPJCL contain the maintenance SYSMODs

CAM42S0	Allocation Manager for older MVS systems
CAM42X0	Allocation Manager MVS/ESA and OS/390
CL142B0	ASM2 Base SYSMODs
CL142I0	IXR SYSMODs
CNS1000	RSVP SYSMODs
CNU4200	ASM2/RSVP common component SYSMODs

   Edit the SYSMOD files deleting any entries for previous maintenance versions applied to your system.

6. Select job MNT2APP, the SMP/E APPLY job. Use the L1CHANGE ISPF edit macro from the install to change job control statement, procedure and zone names. This job uses the same SMPCNTL data sets as those used in the RECEIVE step above. The control files were edited in Step 5 and should be reviewed to make certain they are identical to those used for the RECEIVE step. A return code of 4 is acceptable.

7. Perform product testing to verify that all maintenance has been properly applied. Prior to performing an IPL of the system, make certain the LMP execution key provided on the Key Certificate has been added to the CAIRIM parameters.

8. Select job MNT3ACC - SMP/E ACCEPT job. Use the L1CHANGE ISPF edit macro from the install to change job control statement, procedure and zone names. Edit the MNT3ACC job and make the necessary changes to the *prefix.SAMPJCL* lines.

   MNT3ACC uses the same SMPCNTL data sets as those used in the RECEIVE and APPLY steps above. The control files were edited in Step 5 and should be reviewed to make certain they are identical to those used for the RECEIVE and APPLY steps. A return code of 4 is normal for this job because new modules are being linked to the DLIB in the maintenance.

Appendix A. CA-ASM2 Data Sets

Caution

Do not change the DCB attributes of any of the CA-ASM2 data sets. (Read the following note.)

CA-ASM2 uses the following permanent data sets.

```
CAI.ASM2.ARCH.$DUPPOOL
CAI.ASM2.ARCH.$RAQUEUE
CAI.ASM2.ARCH.$TAPPOOL     CAI.ASM2.BKUP.$TAPPOOL
CAI.ASM2.ARCH.$ARQUEUE
CAI.ASM2.ARCH.ARCLOG*      CAI.ASM2.BKUP.ARCLOG*
CAI.ASM2.ARCH.LOAD
CAI.ASM2.ARCH.LOPERM
CAI.ASM2.ARCH.LOSYS
CAI.ASM2.ARCH.LOUSER       CAI.ASM2.BKUP.LOUSER
CAI.ASM2.ARCH.INCR         CAI.ASM2.BKUP.INCR
                           CAI.ASM2.BKUP.DEFRAG.LOUSER
                           CAI.ASM2.BKUP.DEFRAG.INCR
CAI.ASM2.ARCH.PROTECT
CAI.ASM2.ARCH.SCRATCH
CAI.ASM2.ARCH.SMONFILE
CAI.ASM2.R42.PARMLIB*
CAI.ASM2.R42.SAMPJCL*
CAI.ASM2.ASM2IPC
CAI.ASM2.A2M2JNL
CAI.ASM2.ARCH.RSVP.TABLE
```

Note: You can change the block sizes of the data sets marked with asterisks (*) by specifying BLKSIZE in the JCL used to format them. See the appendix on "Formatting CA-ASM2 Data Sets."

A.1 Usage

This appendix lists data sets used by CA-ASM2 and describes their functions. Remember that you can prefix the indexes ARCH and BKUP with your installation-chosen prefix.

Information on formatting CA-ASM2 data sets is contained in the appendix "Formatting CA-ASM2 Data Sets."

Data Set	**Function**
CAI.ASM2.ARCH.$DUPPOOL	Contains volume serial numbers of tape reels eligible for use as duplicates of master archive tapes.
CAI.ASM2.ARCH.$RAQUEUE	Contains the queue of deferred reload requests. $RXQEXIT controls which reload requests are serviced immediately and which ones are placed into this data set. A periodic batch job (or optionally a queue monitor) scans ARCH.$RAQUEUE at regular intervals. If enough entries are queued, it initiates a retrieval job to process them. As each request is processed, its entry is dequeued from this data set.
CAI.ASM2.ARCH.$TAPPOOL	Contains volume serial numbers of tape reels eligible for use as master archive tapes.
CAI.ASM2.ARCH.$ARQUEUE	Contains the queue of outstanding user requests for data set archival and backup. $AR and $BK are inoperative when this data set is full. A request remains in the queue until the requested data set is archived or backed up, or until the request is found to be unserviceable.
CAI.ASM2.ARCH.ARCLOG	Journals selected CA-ASM2 activity initiated by data center. CA-ASM2 creates a record every time data is archived or backed up from a volume, whenever a tape is copied, when a maintenance request such as $PURGETP is handled, and so forth.
CAI.ASM2.ARCH.INCR	Contains unblocked catalog updates for data sets unloaded in all $ARCHIVE runs. In the event of system crash/operator cancel during catalog update in the UNLOAD 30 step, the retention of the catalog updates in this data set ensures that a subsequent run can successfully merge this data into the IPC as part of the catalog stabilization process.
CAI.ASM2.ARCH.LOPERM	Contains a pointer to the next available slot on tape to which the next PERMANENT ($PRMPASS) user request for archival should be directed. It may contain volsers of tapes awaiting duplexing if the system crashed or the operator canceled $ARCHIVE run between UNLOAD30 and TPBKUP70 steps.
CAI.ASM2.ARCH.LOSYS	Functions identically to ARCH.LOPERM for system-initiated archive runs.

Data Set	Function
CAI.ASM2.ARCH.LOUSER	Functions identically to ARCH.LOPERM for initiated archive runs to nonpermanent tapes. Together with BKUP.LOUSER and BKUP.DEFRAG.LOUSER, the data sets ARCH.LOPERM, ARCH.LOSYS, and ARCH.LOUSER define the current (live) tapes for up to five different classes of archive/backup runs. In all cases, these tapes are five distinct tapes. (See note at the end.)
CAI.ASM2.ARCH.PROTECT	Specifies data sets and the high-level qualifiers of data sets which are never to be scratched by the Controlled Scratch component.
CAI.ASM2.ARCH.SCRATCH	Contains the scratch selection data set which identifies data sets to be scratched by the Controlled Scratch component. It is written to when a $SCRATCH or $PROTECT command is executed.
CAI.ASM2.ARCH.SMONFILE	Contains the Realtime Space Monitor data set which is used to record DASD volume space utilization statistics at user specified intervals.
CAI.ASM2.BKUP.$TAPPOOL	Contains volume serial numbers of tape reels eligible for use as masters and duplicates of backup tapes.
CAI.ASM2.BKUP.ARCLOG	Same function as ARCH.ARCLOG for backed up data sets.
CAI.ASM2.BKUP.INCR	Same function as ARCH.INCR for backed up data sets.
CAI.ASM2.BKUP.LOUSER	Same function as ARCH.LOUSER for $BK requests and system-initiated backup runs.
CAI.ASM2.BKUP.DEFRAG.LOUSER	Same function as BKUP.LOUSER for full-volume ($DEFRAG) backups. (See note at the end.)
CAI.ASM2.BKUP.DEFRAG.INCR	Same function as BKUP.INCR for full-volume ($DEFRAG) backups. (See next note.)
CAI.ASM2.R42.PARMLIB	Contains all of the CA-ASM2 processing parameters including $OPTIONS.
CAI.ASM2.R42.SAMPJCL	Contains all of the sample JCL used during the installation of CA-ASM2 including control cards. There are also several supporting documentation members to assist in the installation of the product.
CAI.ASM2.ASM2IPC	Is the Integrated Product Catalog VSAM key-sequenced data set containing information that prior to Version 4.0 was maintained separately in the archive, backup and IXR catalogs.
CAI.ASM2.ASM2JNL	Is the Journal which serves as a backup to the IPC. The journal contains information needed to recover the IPC if it is destroyed.
CAI.ASM2.ARCH.RSVP.TABLE	Contains the VSAM CA-RSVP sample commands table used by the ISPF CA-RSVP dialogs.

Note: $DEFRAG does not have "current/live" tape that is used in the next run. It always starts with a "new tape" and empty LOUSER file. (It does not check what is there, it just overwrites. It does check INCR and stops if the first character of the record is not a blank.)

Appendix B. Formatting CA-ASM2 Data Sets

This Appendix lists the CA-ASM2 data sets that the $FORMAT utility preformatted during the installation and gives further information on the use of $FORMAT.

B.1 $FORMAT Utility

During installation, the $FORMAT utility preformats the following CA-ASM2 DASD data sets.

DDNAME	Archive Data Set	Backup Data Set	Notes
$LOG	ARCH.ARCLOG	BKUP.ARCLOG	A,D,I
ARCHBKLG	ARCH.$ARQUEUE	BKUP.$RAQUEUE	C,G,J,F
$ASM2INC		BKUP.DEFRAG.INCR	B,C,K
$RXQUEUE	ARCH.$RAQUEUE	....	C,G,H
USERDD	For installation's optional use		A,C,E
LOxxx	ARCH.LOUSERxx ARCH.LOPERMxx ARCH.LOSYSxx	BKUP.LOUSERxx BKUP.LOSYSxxx BKUP.DEFRAG.LOUSERxx	D,L,M

For every run, the JCL must include a SYSPRINT DD statement. You may specify BLKSIZE on the statement, but $FORMAT forces LRECL=125 and RECFM=VBA.

$FORMAT formats each of the CA-ASM2 data sets according to internal requirements.

The Controlled Scratch component also has a preformatted data set, CAI.ASM2.ARCH.PROTECT which is formatted by the utility program ASM2PUUL during installation. The procedures ASM2PUUL and ASM2PCUL are provided in CAI.CAIPROC to perform updates to the protect file after installation. These procedures are described in the *CA-ASM2 System Reference Guide*.

Notes:

A. You can specify BLKSIZE for this file in the JCL. If not specified, it defaults to single blocking. If the block size is not a multiple of LRECL, it reduces to the next lower multiple.

B. LRECL is fixed at 200 for Version 4.2.

C. The first extent only is formatted with as many records as can fit.

D. One formatted record only is written.

E. You may specify LRECL.

F. If you need to expand the Queue Manager files, ARCH.$ARQUEUE and ARCH.$RAQUEUE, or just want to know their format, read the appendix on "Expanding the Queue Manager Files."

G. For the Queue Manager files, the LRECL must be a multiple of 256, and the BLKSIZE must be a multiple of the LRECL. The $FORMAT default for LRECL is 256; and for BLKSIZE 256.

H. You must have the BLKSIZE=LRECL. If not, $FORMAT changes the BLKSIZE to equal the LRECL.

I. RECFM=F,LRECL=144,BLKSIZE=144,DSORG=PS, Must be contiguous.

RECFM=F,LRECL=256,BLKSIZE=256,DSORG=PS, Must be contiguous.

K. RECFM=FBS,LRECL=200,BLKSIZE=200,DSORG=PS, Must be contiguous.

L. LRECL and BLOCKSIZE the same, up to 256 bytes. Only the first record is used.

M. xx, if present, specifies an optional RUNID.

$FORMAT Sample JCL

```
//STEPNAME    EXEC    PGM=$FORMAT
//STEPLIB  DD  ...
//SYSPRINT DD SYSOUT=A
//$ASM2INC DD DISP=OLD,DSN=BKUP.DEFRAG.INCR
//$LOG     DD  DISP=OLD,DSN=ARCH.ARCLOG
//ARCHBKLG DD DISP=OLD,DSN=ARCH.$ARQUEUE
//$RXQUEUE DD DISP=OLD,DSN=ARCH.$RAQUEUE
//LOXXX    DD DISP=OLD,DSN=ARCH.LOUSER
```

Caution

All files whose ddname is recognized are formatted. For a selective formatting, remove or comment out unwanted DD statements.

Expanding CA-ASM2 Data Sets

From time to time it may be necessary to make more space available to the data sets. Simply reallocate the dataset as follows:

1. Choose a time when the data set is empty.
2. Scratch the data set.
3. Run $FORMAT with the DD statement for that data set only, allocating the total number of blocks required in the JCL.
4. Recatalog the data set if necessary.

Appendix C. Expanding the Queue Manager Files

During the CA-ASM2 installation, the $FORMAT program formats each Queue Manager file as a BDAM (Basic Direct-Access Method) data set. ARCH.$ARQUEUE is the queue for archive and backup requests; ARCH.$RAQUEUE is the queue for reload requests. See the appendix on "Formatting CA-ASM2 Data Sets."

Later if you need to expand either Queue Manager file, reallocate the data set with half-track or full-track blocking for best performance. Follow this general procedure:

1. Choose a time when the queue data set is empty.
2. Scratch the data set.
3. Run $FORMAT with the DD statement for that data set only, allocating the total space required in the JCL.

   a. If an LRECL is not specified, $FORMAT defaults it to 256. If an LRECL is specified, $FORMAT forces it to some multiple of 256 bytes, either the value you choose or the next lower multiple of 256.

   b. If a BLKSIZE is not specified, $FORMAT defaults the BLKSIZE as if 256 were specified. $FORMAT forces the BLKSIZE to some multiple of the LRECL, either the value you specified/defaulted or the next lower multiple of the LRECL.

   c. The LRECL value determines the maximum number of queue records that could possibly be put in the file. If more tracks or cylinders are allocated than can be used, $FORMAT frees up the excess tracks or cylinders.

4. Recatalog the data set, if necessary.

In either Queue Manager data set, each record except the first is divided into 256-byte queue slots. The first LRECL bytes of the first block contain the queue-slot bit map. The first two bytes of this block tell the number of succeeding records (the total number of records minus one). The next LRECL bytes (-1 of the first block) are the map of the queue slots contained in the other records. Each bit in the map corresponds to one slot. If the bit is on, the slot is in use. If the bit is off, the slot is not in use.

You can calculate the maximum number of queue slots possible in the data set two ways. The first way gives the theoretical maximum, based on the chosen LRECL and on the structure of the bit map. Since each bit in the map corresponds to one queue slot, the number of slots cannot exceed the number of bits available for the map. The number of bits in the bit map is eight times the number of bytes. From this, subtract the 16 bits (two bytes) used for the block count that precedes the bit map:

```
Theoretical max. = (BLKSIZE X 8) - 16
```

The second calculation, the actual maximum, is based on both the block size and the number of blocks allocated. It is simply the number of blocks available for the queue, times the number of queue slots per block:

```
Actual max. = (Records allocated - 1) X (BLKSIZE / 256)
```

One is subtracted from the number of records allocated because the first record contains the block count and bit map.

$FORMAT formats the queue data set so that the number of queue records does not exceed the smaller of these two values.

The following example helps to clarify this: ARCH.$ARQUEUE is being put on a 3380 and must be able to contain 8000 commands. Using the default BLKSIZE=6144, you need to find the LRECL to use and the number of tracks to allocate.

1. The LRECL must be a multiple of 256. The first 256 bytes map ((256 X 8) - 16) = 2032 records. Each additional 256 bytes map an additional 2048 records. Therefore, for 8000 records, you need a BLKSIZE = 4 X 256, or BLKSIZE=1024.
2. Each 3380 track holds 7 blocks with BLKSIZE=6144. Each block contains 6144/256 = 24 queue slots. Therefore, the first tracks contain 6 blocks X 24 slots per block, or 144 slots.

Note: The first record is reserved for the bit map. Each additional track contains 7 blocks X 24 slots per block, or 168 slots.

The number of 3380 tracks needed is:

```
    8000 slots - 144 on first tracks
    --------------------------------   + 1 = 48
        168 per additional tracks
```

If you use default block size of 6144, 3380s hold 7 blocks per track, and 3350s hold 3 blocks per track.

Appendix D. MLPA USERMOD Techniques

The MLPA (Modified Link Pack Area) testing technique offers users a means of testing system modifications to LPA modules with minimal risk. CA recommends that sensitive USERMODs which may not be installed with CAIRIM be initially installed using the MLPA technique. After the USERMOD has been verified to be working properly, it may be moved to SYS1.LPALIB.

The following guidelines are offered to assist you in installing a CA-ASM2 system modification using the MLPA technique in conjunction with SMP/E. This technique has the benefit of having SMP/E control the construction of the system module, while isolating the changes in an MLPA data set.

1. Identify which load module contains the module or CSECT (expressed here simply as CSECT even though the module may contain multiple CSECTs) which you are about to modify. You must also identify all aliases of this load module. To identify the load module, run an SMP/E LIST MOD(CSECT name) against the operating system target zone. This identifies the LMOD (load module) which contains the module you are modifying. Next run an SMP/E LIST LMOD(lmod name) against the load module name just returned. This identifies all aliases associated with the load module. If a load module has many aliases, allocate a disk data set to contain the output of the SMPLIST DD report. This data set may then be edited to obtain the aliases without having to key them in, possibly introducing errors.

2. Select the MLPA data set you are using. This may be SYS1.SVCLIB, SYS1.LPALIB or a data set in the LNKLST concatenation. You should use either SYS1.SVCLIB or a LNKLST data set to prevent changes from going into SYS1.LPALIB until the USERMOD is verified. This prevents the changes from being installed prematurely in the case that an IPL with CLPA is unintentionally performed.

3. Construct and run an IEBCOPY job to copy the load module and all aliases from SYS1.LPALIB to the MLPA data set selected. (Prior to running this copy job, ensure that no other copies of this module are in other MLPA data sets.) When the SMP/E USERMOD is constructed, the MLPA data set is used to override the LPALIB DD in the operating system SMP/E procedure. Because SMPx "knows" that the load module already exists in LPALIB, it includes the old load module after including the modified CSECT to obtain all other CSECTs in the load module.

4. Modify the SMP/E USERMOD provided to add an override for the LPALIB DD statement specifying your MLPA data set. Check the order of the DD statements

in your MVS SMP/E procedure to ensure that the procedure override is performed correctly. If the overrides are not ordered correctly you may end up modifying the module in SYS1.LPALIB instead of your MLPA data set. When you execute the SMP/E USERMOD to install the system modification, SMP/E links the new load module to the MLPA data set. The SMP/E target zone is updated to show the USERMOD as applied to the data set associated with the LPALIB DD, however the MLPA data set has been updated instead of SYS1.LPALIB.

5. Update the SYS1.PARMLIB IEALPAxx member to include the load module and all aliases from the correct MLPA data set. If the MLPA data set is in the LNKLST concatenation, you need only specify SYS1.LINKLIB for the data set name, not the actual MLPA data set name. (MVS/ESA 4.1 changes this convention; at this level the MLPA data set name is actually specified in a new command syntax.) You should also verify the MLPA= parameter in the IEASYSxx member and the LNKLSTxx member if adding a new MLPA data set to the LNKLST. Refer to the appropriate "MVS System Programming Library: Initialization and Tuning Guide" for your system for further details on the SYS1.PARMLIB members.

6. IPL your system using the MLPA data set and verify that the system modification operates correctly.

7. When the system modification has been verified, it may be moved to SYS1.LPALIB in one of two ways: You may either copy the load module and all aliases from the MLPA data set to SYS1.LPALIB replacing the system versions (SMP/E has already been updated to reflect the changes to SYS1.LPALIB), or you may remove the LPALIB DD statement override and execute SMP/E to apply the USERMOD over again specifying the REDO parameter on the APPLY.

8. Update the IEALPAxx member to remove the module and all aliases which you have just moved to SYS1.LPALIB.

Appendix E. Aging Conversion Utility

A new version of the IFG0196W OPEN modification was provided starting with Version 4.0 of CA-ASM2. This OPEN modification saves CA-ASM2 aging information at new offsets in the Format-1 DSCB which are compatible with recent changes to the Format-1 DSCB required by SMS (DFP 2.4 and above). Current CA-ASM2 users with the IFG0196W OPEN modification installed (still using the old offset of 80 for LMTIM) are strongly encouraged to install the new version of IFG0196W due to the current maintenance levels of IBM products. Using the old offset for flag bits may cause unpredictable results.

As a part of implementing the new IFG0196W module, users must execute the Aging Conversion Utility, ASM2IAGE, to modify the aging information in the Format-1 records of the VTOCs saved by previous versions of CA-ASM2. CA-ASM2 clients may execute ASM2IAGE to either:

- Remove all CA-ASM2 aging information from the Format-1 DSCB.
- Move the CA-ASM2 aging information to SMS compatible locations in the Format-1 DSCB.

Users who are installing the CA-ASM2 OPEN modification for the first time do not need to run ASM2IAGE. Execution of the ASM2IAGE should be scheduled after Version 4.2 of the OPEN modification IFG0196W has been installed.

E.1 Conversion Requirements

The ASM2IAGE utility is installed in CAI.CAILIB when the CA-ASM2 base component is applied with SMP/E. If your installation has its own special Format-1 DSCB modifications, the source code for ASM2IAGE is provided in member ASM2IAGE of CAI.CAISRC. The JCL to assemble ASM2IAGE is provided in member CONVASM of SAMPJCL.

To convert the Format-1 DSCBs, execute the following JCL distributed as member ASM2CONV of SAMPJCL.

```
//AGECONV  JOB (ACCT INFO),CLASS=A,TIME=30,MSGCLASS=A
//*
//ASM2CONV PROC
//ASM2CONV EXEC PGM=ASM2IAGE,PARM=SMSCONV,REGION=4096K
//STEPLIB  DD   DISP=SHR,DSN=CAI.CAILIB                     See Note 1
//ASM2RPT  DD   SYSOUT=(*)
//SYSUDUMP DD   SYSOUT=(*)
//AVxxxxxx DD   DISP=SHR,UNIT=SYSDA,VOL=SER=xxxxxx          See Note 2
//         PEND
//*
//M2CONV   EXEC ASM2CONV
//
```

Note 1 Alter the STEPLIB DD statement to point to the load library that contains your CA-ASM2 load modules.

Note 2 Each disk volume that has its Format-1 DSCBs converted needs an AVxxxxxx DD statement. The DD statement that starts with AVxxxxxx should be altered to point to the VOLSER of the disk volume that you wish to run the CA-ASM2 conversion against.

If you wish to run the conversion program against PACK00, PACK01 and PACK02, the AVxxxxxx DD statements would look like:

```
//AVPACK00   DD  DISP=SHR,UNIT=SYSDA,VOLSER=PACK00
//AVPACK01   DD  DISP=SHR,UNIT=SYSDA,VOLSER=PACK01
//AVPACK02   DD  DISP=SHR,UNIT=SYSDA,VOLSER=PACK02
```

The conversion program has two modes of operation. One mode must be chosen before executing the program. The two modes are:

- Specify PARM=SMSCONV on the execute statement to have the CA-ASM2 last modified time field moved to the new location in the Format-1 DSCBs.
- Specify PARM=UNINIT to uninitialize or completely backout all CA-ASM2 aging information from the Format-1 DSCBs. This should only be used if you wish to completely remove all aging data from your Format-1 DSCBs.

All disk volumes do not have to be converted in a single run. It is suggested that the disk volumes be converted in strings; a string at a time.

Note: While the conversion program is processing the Format-1 DSCBs on a pack, a RESERVE is issued against the VTOC of the disk volume. Once the conversion of the Format-1 DSCBs is complete on a disk volume, the RESERVE is released.

The conversion program may be executed multiple times against a disk volume because only unconverted Format-1 DSCBs are converted.

Once the conversion program has successfully completed, you notice that CA-ASM2's last updated time field is at displacement X'47' for 2 bytes in the Format-1 DSCB and the 4 bytes starting at X'4E' are zeros. If you have any in-house utilities that are expecting CA-ASM2's last updated time field to be at displacement X'50' for 2 bytes they need to be updated to pick up this field at the new location.

E.1.1 Functional Considerations

There are some functional considerations that should be kept in mind after you have performed your conversion of the Format-1 DSCBs.

1. If you have decided not to convert all of your CA-ASM2 aged Format-1 DSCBs at one time, you should be aware of the consequences of this in regards to executing $MIGRATE and $DEFRAG.

   When executing $MIGRATE with the DSCBCOPY parameter, you should not migrate a data set that exists on a non-converted disk volume to a converted disk volume because $MIGRATE simply copies the unconverted Format-1 DSCB to your converted VTOC. Executing $DEFRAG to reload to an unconverted disk volume reloads the Format-1 DSCBs back down in the format they were backed up in. Thus, if you have run a $DEFRAG against an unconverted disk volume and then convert the disk volume to the new CA-ASM2 format and then issue a full-volume or selective reload from the old $DEFRAG tape you cause unconverted Format-1 DSCBs to be reloaded down to your converted disk volume.

   Note: This does not occur when executing $RA and $RB commands, unless your site has chosen to save this Format-1 DSCB information when an ARCHIVE or BACKUP is performed. For more information on this topic see Item 4.

2. Executing utilities which restore Format-1 DSCBs that were created prior to converting to the new CA-ASM2 format causes old Format-1 DSCBs to be reloaded onto the converted disk volumes. The vendors of these utilities should be contacted to obtain PTFs to keep the utilities from restoring the data at X'50' - X'51' in the Format-1 DSCBs.

3. If you happen to mix in some old Format-1 DSCBs onto a converted disk volume you can either:

   - Wait until the data sets in question are next opened and at that time CA-ASM2's OPEN modification dynamically recognizes an old Format-1 DSCB and reformat it into the new format; or

   - As an alternative, rerun the ASM2IAGE Format-1 DSCB conversion program against the disk volume in question.

4. If CA-ASM2 is restoring the last modified time in the Format-1 DSCB during a RELOAD request by utilizing the $UDSCB in $OPTIONS, you have to modify it to point to the new displacement of the last modified time field. The last modified time field was located at displacement 80 (X'50') for 2 bytes in the Format-1 DSCB. Therefore, if the current setting of $UDSCB includes the displacement of 36 (this is the displacement from the Format-1 identifier for the old location of the last modified time field), the $UDSCB should be altered to the following to pick up the new location:

```
$UDSCB   DS   0X
              DC   AL1(27)   Displacement from
                             Format-1 identifier.
              DC   AL1(2)    2 bytes in length
```

5. If you have multiple CPUs, these procedures must be implemented in all environments.

Index